TURING

New Riders



瞬间之美

Web界面设计如何让用户心动

[美] Robert Hoekman, Jr. 著

向怡宁 译



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目（CIP）数据

瞬间之美：Web 界面设计如何让用户心动 /（美）霍克曼（Hoekman, R.）著；向怡宁译 . —北京：人民邮电出版社，2009.7
书名原文：Designing the Moment：Web Interface Design Concepts in Action
ISBN 978-7-115-20767-8

Ⅰ. 瞬… Ⅱ. ①霍…②向… Ⅲ. 主页制作 – 程序设计 Ⅳ. TP393. 092

中国版本图书馆CIP数据核字（2009）第060622号

内 容 提 要

本书是界面设计领域的畅销书中文版。书中通过 30 多个故事，阐明了如何将好的设计原则运用到实际的 Web 应用程序界面中，使界面引人入胜，如何创造完美的瞬间，给用户以美好的体验。

本书适用于各层次 Web 设计人员和开发人员使用。

瞬间之美：Web界面设计如何让用户心动

◆ 著 [美] Robert Hoekman, Jr.
译 向怡宁
责任编辑 傅志红
执行编辑 杨 爽 印星星

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京 印刷

◆ 开本：880×1230 1/32
印张：7.625
字数：235千字 2009年7月第1版
印数：1 – 3 000册 2009年7月北京第1次印刷
著作权合同登记号 图字：01-2008-5820号

ISBN 978-7-115-20767-8/TP

定价：45.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223

反盗版热线：(010)67171154

译　者　序

随着 Web 2.0 掀起的迅猛风潮，互联网在今天已经发生了极大的改变。旧日的大小信息网站有如昨日黄花，显得不合时宜，以前由少数人编辑或定制内容的互联网早已变得更为开放和多元化，以前的“只读式互联网”变成了“可读可写的互联网”。我们身边的每个人都能够为其添加内容，贡献自己的力量。我们发布自己的博客，同时还浏览其他人的博客。如果有想说的话，我们还会发表评论或回复。我们为各种文章标注 Tag 标签进行分类和归总，既方便了自己，又方便了他人。我们点击查看别人的照片，同时也上传自己的照片。我们订阅各种感兴趣的 RSS。我们使用 Google 或者百度就好像在图书馆里自由地检索。我们使用维基百科，而且只要愿意也有自信，还可以为其添砖加瓦。我们在今天的互联网上学习新技术、欣赏流媒体视频、交流读书或观影的感想、结交彼此倾慕的朋友。这是一个史无前例的“共享”的世界。

没错，史无前例。在这个也许几年前根本没人能够想象的平台上，作为产品开发者或设计者的我们，应该遵循怎样的标准才能在最终交出一份完美的答卷呢？你知道，完美的答卷就意味着以万为单位的点击量、良好的客户口碑，以及（最重要的）丰厚的市场回报。

我想你已经知道，Web 2.0 的主角是它的使用者，也就是我们所说的用户。用户才是所有网站或者 Web 应用的评判者（当然，也是财源）。用户认为我们提供的服务物有所值，他们才会口耳相传，才会把心得写到自己的博客里，才会像苹果教徒[①]那样成为你的拥护者。用户从不会手拿标准答案来为我们的产品打分，他们依据的是自己在你的网站或 Web 应用中，每一次发出指令或得到反馈的瞬间体验。如果他们在完成任务时

①“苹果教徒”特指那些对 Apple 公司具极高品牌忠诚度的用户。这种人对 Apple 的产品不仅只是单纯地使用，而且变成了一种爱好，其激情甚至会演变为着魔的状态。表现在疯狂搜集相关的各种产品、纪念品，在新产品发布前数天就在零售店前排队，甚至用 Apple 公司的 logo 进行纹身等。——译者注

感觉不到麻烦，那么你的得分就是“良好”；如果他们在此基础上还能感到成功的欣喜或者操作的愉悦，那么评分就是“优秀”。

所以这本书开篇的第一句话就是：

良好的用户体验，全在于那些完美的瞬间。

Robert Hoekman, Jr. 是一名交互设计师和易用性专家。他曾任职于 GoDaddy.com、Macromedia、Adobe 等多家知名公司，现在有了自己的工作室 Miskeeto，一直致力于为广大受众提供优质的用户体验服务。之前他所著的 *Designing the Obvious*（国内译版名为《一目了然》）直观地向我们阐述了如何构建简洁易用的软件显性设计方法，而这次他又带着本书的英文版 *Designing the Moment* 再次揭示关于“瞬间”的秘密。

用户身处我们的网站或 Web 应用中，由始至终要经历一系列瞬间。不管是了解新特性、注册账户、评论及评分，还是搜索信息、填写表单或者订阅 RSS，甚至在他们取消账户准备一去不返的那些时刻，所有这些形成一个又一个瞬间的体验。Robert 根据自己的亲身经历和设计经验将其归结为 30 多个故事，基本上涵盖了所有可能出现的瞬间。每一个瞬间就是一个故事，每一个故事的结尾都有一个最合适的解决方案。Robert 从不讲大道理，他给我们的是最简单、最直接的礼物——如果你从没研究过用户体验，遇见问题时按他说的方法去做就不会出岔子；如果你算是半个专家，书中字里行间流露的智慧和思考方式也会让你受益匪浅。

而且，我们很高兴地看到他并没有刻意地美化自己，而是把每次失败后探索并改进的全部过程都展现出来，甚至还有如何“剽窃”其他优秀设计的手段。首先，再知名的设计大师也会出错；其次，我们能领悟什么叫做“借鉴”而不是“抄袭”；而最重要的，我们将从中学习到如何发现设计的不足并持续改进的方法。

感谢图灵公司对我的邀请，翻译这本书的过程也可以说是一次学习的过程。其中有些章节所讲述的内容直接用在了我手头的项目中。而且 Robert 的旁征博引也让我开阔了眼界，了解了很多以前尚未接触的东西，而它们都非常有意思。所以我真的很希望你们都能读到它，相信你们会和我一样，翻完最后一页时将切实感受到自己又迈上了一级新的台阶。

本人才疏学浅，尽管慎之又慎，译文仍难免有疏漏之处。恳请大家指正，希望大家共同交流进步：messiah.book@gmail.com。

向怡宁
2008 年 11 月于北京

前　　言

良好的用户体验，全在于那些完美的瞬间。

在第一个瞬间，也就是当一位用户从 Google 搜索结果页面跳转到某个网站的时候，他此刻可能是想看看这个网站的名字及其功能；而下一个瞬间可能就开始琢磨如何使用页面中的播放器来观看某个视频介绍。再然后则可能是寻找从哪里注册，或者看看商品是否打折，或者尝试联系这个公司。

很多设计师总喜欢从“标准用户（Persona）”（即针对某类产品设想的典型用户）身上捕捉那些一般性意图。但上述这些目标不能算是生活体验，而更应算作交互式体验目标。

若想完成优秀的设计，我们就需要时刻自问：在这个瞬间我们希望用户做什么，界面应该如何鼓励他们完成这个任务？

而且还需要考虑这位用户在此刻的目标是什么？我们的设计又如何帮助他实现他的目标？他的目标是想了解一个新网站吗？是想找到某些具体的信息？是想填一个表格？还是想往购物车里面塞点什么？

每一个瞬间都有可能加强也可能摧毁用户对某个产品或公司的信心。每一个瞬间对于完整的用户体验来说，都是不可或缺的一部分。

何出此言？因为每个人在每个时刻所要完成的任务，对于此时的他而言，那就是最重要的任务。

确保每时每刻都完美，这是设计师的职责。要确保我们的用户在这个瞬间心情愉快，效率很高，从而自我感觉得心应手。

我们的工作就是考虑所有这些瞬间——设计一些东西来支持用户达成每一个目标，而不会干扰或者妨碍他们。同时，把这些完全不同的部分凝聚为一个整体。

我们的工作不是设计花哨的屏幕，而是要设计美好的瞬间。

无论是简单还是复杂的应用都是如此。最重要的是，当用户想要达到其交互目标的种种时刻，都会发生些什么。

要实现优秀设计的关键之一，就在于了解每一个瞬间需要发生一些什么才能完美。而我们就应该去实现它们。

为瞬间而导航

在每一个瞬间，用户都将发出各种各样的行为。他们在一天之内可以从 Amazon 游弋到 Z Gallerie[①]，然后可能做出从注册到退出之间的任何事情。

用户在互联网上的行为多种多样，有输入，有编辑，有搜索，有调整版面，有创建，有上传，有删除，有共享，有组织，有参与。这些都是设计师希望用户进行的行为。在大多数情况下，这也是用户正想要做的行为。

当然，我们也会做一些通常自己并不会意识到的事情。我们分析，我们判断，我们遗忘、犯错误、走神、改变主意、迷失方向、感到困扰。而如果幸运的话，我们会学到东西，得到指引。我们会有新的想法、形成记忆、熟悉、信任、灵感乍现，并且觉得自己的工作富有成效。

正是这些行为才让我们成为活生生的人。这本书就是要讨论如何设计那些支持所有此类行为的界面——在用户需要进行选择的那些瞬间，帮助他们做出正确的决定。我们将讨论创建怎样的交互行为以鼓励人们输入、编辑、搜索和共享，激励他们完成所有我们希望他们做的事。我们还将讨论如何设计各种手段，来帮助人们提升工作效率——即使他们天生忘性大、老出错，或者总是三心二意。

更确切地说，这本书收集了 30 多个故事，阐明了如何将好的设计原则运用到实际的 Web 应用程序界面中去，使界面引人入胜，呼之欲出。自始至终，这些故事都将以批判的眼光来看待设计，并且详究每一个细节，力求确保人类——这种时常犯错而又行事乖张的生物——能不受软件产品的限制，而又自我感觉良好。

要设计美好的瞬间，就需要做到以下几点。

---

① Z Gallerie 是一个主营家居装饰的国际电子商务网站。网址为 www.zgallerie.com/。此处作者意指用户从 A 逛到 Z。——译者注

- 介绍一种具启迪意义并且富有洞察力的界面设计方法，我称之为“大声想出来”；
- 以批判的眼光遍查用户与 Web 应用交互的每一个阶段中的每一个元素，一次就看一个瞬间；
- 为所有的设计——从页面布局到在线社区——推荐一整套最优方案。

最重要的是，本书探究了界面的种种微妙细节，它们在用户使用的每一个瞬间如何改善或是破坏用户的体验，进而探讨了应该如何对每一个细节加以改进。

交互的设计

本书的结构框架遵循了用户进行 Web 应用操作时的典型步骤，也就是从一开始最关键的“第一印象”，一直到最后他们退出并转向别处。每个部分的标题直接与这些行为相关，每部分都讲解在这一个瞬间用户实实在在要做的事情。

第一部分到第三部分讲述的是当用户打开一个新网页后在头 30 秒钟内所要做的事情。这些章节里面的故事描述了他们首次了解界面、探寻方向、略读文本熟悉更多情况、观看视频和动画演示，以及搜索详细信息等情形。

第四部分到第六部分则是关于用户在深入我们的网站时所做的事。一旦他们决定要创建自己的账号，就会输入信息、编辑内容、管理数据，以及通过种种方式与网站进行交互。在这个过程中，每一个瞬间都有可能持久地影响他们对网站的信心和好感。

第七部分则关注当用户离开网站——不管是暂时离开还是永久离开——时所要发生的事情。

换句话说，我有意地将这整本书的结构做如此安排。其目的是使它能重现出当用户面对 Web 应用时由始至终的所有细节。我们将详细地探讨人机对话中的每一个阶段，不管它是只有 3 秒钟还是将会持续两年。如果用一句话来概括，那就是在这些故事中所讨论的元素将会囊括在线用户体验的完整情境。

这样的顺序安排也方便你在日后能够随时翻阅，把这本书当作某种“灵感参考指南”来使用。如果你在大半年后为了某个编辑功能的设计而一筹莫展，可以打开书里相应的章节（第五部分）读几页，很有可能会得到不少启发。

而这些都是源于“讲故事”这种永不过时的艺术。

来自真实世界的问候

在这些故事里，有一些讲述了如何贯彻和实施那些常见的设计原则，还有些则会探讨你可能从未遇见过的个别情况。有一些故事提出了新的见解和看法，还有些则似乎是老生常谈。不论怎样，它们存在的目的都是为了启发你的思维，揭示各种事实和发现，提出理论培养直觉，最终让你能设计出各个伟大的瞬间。

在 *Designing the Obvious*（之前我写的一本书）①中，我详细地阐述了关于 Web 应用设计的七个核心原则。在此基础上形成了一套通用的方式方法，能够举一反三、成功地应用到任何 Web 项目中去。

而在本书中，我用一系列故事来演示我是如何把这些想法和观点应用到实际项目中的，从而为用户创造出令其印象深刻的瞬间。与此同时，我还继续提出了大量新的想法和观点。

这些故事有些是我的亲身体验，是我日常在进行设计时碰到的问题，而有些则是全新的故事，是我专门为了本书而设计的。

所有的故事都直接源自那些痴迷于几乎任何交互式设计的设计师们头脑中的想法。他们当中有些人在深夜辗转反侧地思索当地的快餐厅应当如何摆放饮料机才能方便消费者，还有些人则花费大量时间揣摩那些获取用户体验的各种瞬间，想方设法地加以改进。

大声地做出决定

在 *Designing the Obvious* 的末尾，我谈到了勇敢做出决定的重要性——哪怕明知这些决定在以后肯定会改变。不论决定的有效性能持续多长时间，

① *Designing the Obvious* 由 New Riders 出版社于 2006 年 10 月出版。网址为 http://rhjr.net/dto/。该书的中文版由机械工业出版社于 2008 年 1 月出版，书名《一目了然》。——译者注

都必须做出决定，使这个设计中的所有地方都按此决定来考虑，并且有意为之。决不能听天由命。

在本书中，我尝试做一些之前从没做过的事情：把做出那些决定的详细过程全部揭晓并且逐步剖析给大家。

我希望能揭示出“持续改进”这一设计方法的重要性。我将向你们展示绝对需要采用迭代的方法才能达到好的结果。我将向你们展示如何应用设计原则、探索、经验、直觉甚至感觉来指导设计过程。

换句话说，我大声地做出所有决定。

我承认自己犯过错误，也从其他人那里学到很多，我向你们展示好的设计其实是一大堆差的设计最终进化的结果。希望通过这样做，你们将发现好的设计绝不是单纯因为天才的创意或者灵感爆发的一瞬间而产生的，它是持续前进的永恒动力所造就的产物。

多年之后你也可以重新翻阅这本书，并且质疑我所做的这些决定。可能那时你会发现我在这里所说的一切对你来说都幼稚得可笑，而自己早已经超越了我所在的高度。让这一切在你身上发生，是我最大的心愿。

我把自己敞开给你，是希望能帮助你学习“如何质疑自己”：在每一个设计中找到缺点，并且不断地寻找方案来改进它。

开始交流

这本书决不是一本所谓 Web 界面的权威性指南读物。它只是一个交流的开始，其目的是让你思考。

书里所说的都不是什么权威或者官方的答案，因为本来就没有什么权威或者官方的答案。当我向你具体推荐什么的时候，那只是基于我自身的经验、观点和认识。甚至有时候，它们仅仅只是基于我的直觉（直觉是每一位设计师都应当具有并且相信的东西）。

因此，我几乎可以保证，你将会发现某些地方你能做得更好——你肯定会找到一些我没想到的办法来让你的用户拥有更美好的瞬间。如果你做到了，我希望你说出来。不要做一个整天躺在靠背椅里面的设计师——走出去，告诉人们你所做的改进。

把你的意见和想法发邮件告诉我。把它们放在你的博客里面去。把它们

告诉你的朋友。最好也告诉你的同事，还有你的老板。

我从不认为自己是这个星球上知识最渊博的设计师，也从没相信过自己做的一切都是金科玉律。一直以来我都是从其他人身上学习并获得灵感。通过这本书，我希望向你传授知识并激发你的灵感，但同时也强烈地相信和希望，最终你也能像我一样地传授给其他人。

如果你有什么好的想法和意见，请告诉大家。让交流继续。

即使已经尽了最大的努力，仍然是没有最好，只有更好。书中提到的这些设计我已经尽了自己最大的努力，但如果你发现有什么方法能更进一步，请大声地告诉大家！

因此你也不必做笔记

输入在书里找到的那些 URL 网址链接真的是一件非常繁琐的事。你需要一个字母一个字母地边看边输入。为了避免这个讨厌的过程，我为书里面出现的所有网站、Web 应用程序、博客回复、文章、研究论文和其他参考资料创建了一个存档。在书里你不会看到一个 URL 网址链接[①]。如果你想在那里进行深入研究，请访问网站 https://rhoekmanjr.backpackit.com/pub/1828580-designing-the-moment-links-from-the-book，点击“Links from the book”链接。

不需要再啰嗦什么了，让我们开始吧。

① 为了便于读者查阅，本书以脚注形式给出了所有引用的链接地址。——译者注

目　　录

第一部分　指引用户

第二部分　学　　习

第三部分 搜 索

第四部分 深 入

第五部分　参　　与

第六部分　管理信息

第七部分 离 开

第一部分

指 引 用 户

在互联网上探索，毫无浪漫可言。我们不是 Lewis 和 Clark[①]，把自己的所有家当打完包然后一头扎入西部去发现美国未知的领土。不要幻想在网络中能找到什么尘封的宝藏。而且除了经常使用的 Google 和电子邮件之外，你的大胆探索一般来说也不会获得什么丰厚的回报。

探索万维网通常都带有一定的功利心，就好像你第一次光临某家新店铺一样。尽管曾经逛过上千家店铺，但是我们仍然会觉得新店铺和之前的那些有所区别，因此我们需要先熟悉熟悉这个新的环境。

与此相似，当我们面对一个并不熟悉的网页时，通常都会先扫视一下整个页面，试图了解这个网站大概的用途，然后再决定自己是否应该更深入地浏览，点开网站其他的页面去四处逛逛。

这样一个“着陆”的过程，对于树立用户对网站的信心以及引导用户进行深入探索，是非常重要的。

那么，作为 Web 设计师，我们的工作就是要帮助用户“着陆”，而且越快越好。出于这个原因，本书最开始几章首先讨论用户第一次访问某个站点的头 30 秒钟内将会发生的事情。

在这片刻之内，我们需要完成以下几件事：

- 设计吸引人的第一印象；
- 以明确的路线来提供信息；
- 提供工具指引用户；
- 把观光客转变为消费者。

所以在头几个故事中，我们将巡视从页面布局到导航、从搜索结果到视频演示的方方面面，来看看所有这些元素究竟如何帮助我们达到目的，同时让用户尽快地适应我们的网页。

---

① Lewis 和 Clark 是美国探险家。他们于 19 世纪初进行了历时三年的“Lewis 和 Clark 探险”（Lewis & Clark Expedition），深入西部荒野。这也是美国人第一次横跨北美大陆，在美国西部开发史上意义深远。——译者注

第 1 章
令人心动的第一印象

人们常说第一印象往往会给人留下一生的记忆。对于一个网站而言，这一说法也同样成立。但遗憾的是注意到这一点的站点并不是很多。

有人说 Steve Jobs，这位 Apple 公司无畏的领导者，对第一印象非常执著。也许这可以解释为什么 Apple 公司的产品总能在第一时间就吸引人们的眼球。

哪怕是打开一台还在包装盒里的崭新的 MacBook Pro，就足以让人流口水。首先，单凭外包装就令人满怀期望和惊喜。而一旦打开包装盒，将 MacBook 安置在书桌上，它的启动过程从第一个瞬间开始就极富魅力。伴随着栩栩如生的动画，它用多种语言向你表示欢迎，会自动搜索网络连接，提示你把之前 Mac 电脑中所有的数据和存档备份到新电脑中来，还让你在一片端庄雅致的桌面背景之上进行设置，一切都为用户考虑得细致入微，让这整个过程毫不费力。

要把这一切搬到互联网上则并不容易。不过，设计一个引人注目、令人动心的页面仍然是我们工作中不可缺少的一部分。

前不久有一位客户来找我，带着一大堆关于首页设计的需求。他在设计一个专门面向平面设计师的社区网站。他已经完成了一项困难的工作，就是精简了需求列表，只保留了那些真正有用、确实与网站用户相关的部分。而我所要做的就是把这份列表转变成美好的第一印象。

由于这个网站主要针对平面设计师，因此它的首页需要一个中心区域突出展示大量富有特色的图片，同时还需要一些小区域来安排用户的账号信息、网站特色项目以及登录框。当然，页面中还需要包括那些常见元素，如网站 logo、欢迎辞、导航栏、搜索框以及页脚信息等。

另外，就算这些相对来说尚可应付，网站首页还需要放置一些关于业界

和站点本身的头条新闻以及摘录片段。

仅展示图片一项就将占据很大一块的页面空间。要将其他所有栏目都加进来，同时还要包括新闻区域，就显得有些棘手了。别忘了，即使首页的所有这些部分都能完美地展现，能让用户得到他们想要的东西，但毕竟这个站点主要的业务目的是吸引浏览者们注册并成为该社区的成员。因此我需要一种设计诱使用户注意到 Sign Up Now（现在就注册）按钮，同时让他们想要去单击它。

1.1 发现页面布局

我一开始努力尝试去发现页面的布局。对于页面设计而言，布局是其中最基础的元素。

我在这里用到了“发现”这个词，因为页面的布局设计并不一定出自于设计师本身。当我想到页面布局的时候，经常会联想到 Bob Ross。这个画家最让人津津乐道的一点，就是他能将随意洒到画布上的颜料加工成一副满是“快乐的树”的风景画[①]。他总是根据绘图的进展随机应变，决定下一步该做什么。

明白了吧，不是你选择页面布局，而是它选择了你。

布局是内容组织的必然结果，它往往是自己逐渐成形的。一旦周全考虑了页面的所有需求，并且确定了每一块的内容，你的工作就变成了看着所想要的布局自己展现在你眼前。这就好像你不断地切削石膏直到雕塑成型。

在开始安排页面内容时，我会秉承这一最基本的（同时也相当复杂的）设计方法。简单来说，我把所有的栏目区域都扔到显示屏上，然后移来移去不断地调整让它们彼此协调。

这个过程有点类似玩拼图游戏，只不过没有盒子封面那张最终完成的图作提示，而且拼图的每一个部分都能任意改变大小。

---

① Bob Ross（1942—1995）是美国当代最富盛名的自然主义绘画大师。他出神入化的技巧、行云流水式的画风，开创了绘画的新纪元。他曾经主持了美国最广为人知的艺术类电视节目“绘画的乐趣”，教观众如何绘制“快乐的树”、“快乐的云彩”和“快乐的群山”。在节目每一集的末尾他都会说：“绘画时没人会犯错误，那只是一些快乐的小意外。”——译者注

我最先考虑的是主要图片展示部分，因为它需要的区域最大。与此同时我也颇为得意地创建了一个宽宽的区域放置新闻，因为它需要空间显示文章的摘录片段。至于 My account（我的账户）、Navigation（导航栏）以及 Featured item（网站特色项目），我把它们平等对待，放在了页面左侧。

几分钟之后，我得到了第一个版本，见图 1-1。

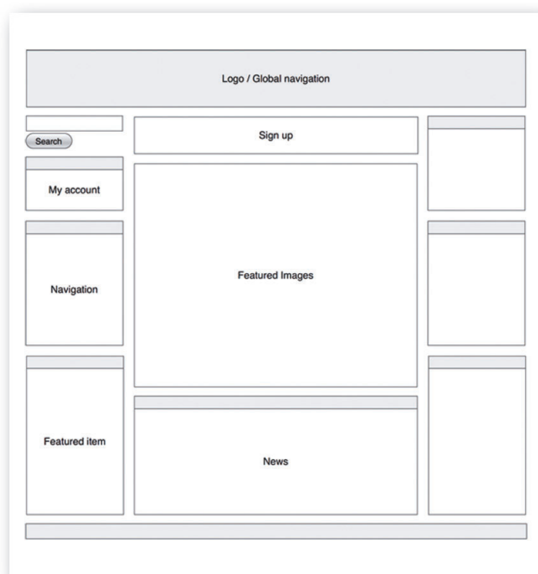


图　1-1

这时我突然感觉到新闻部分似乎太宽了。这个宽度能让我显示很长的头条，也能放很多摘录内容，但同时它也不便于继续划分区域，分别去显示关于站点的内容和关于业界的内容。为了纠正这一点，我把该区域分开为两列，见图 1-2。

好多了。

随后我又意识到另外一个问题。刚开始设计时，我猜测自己可能需要 3 列来放置所有的内容。而现在发现完全不需要那么多栏目就能够解决问题，因此我移去了第三列。然后我加大了左侧列的宽度，这样能为站点导航栏和网站工具列表留下更多的生存空间。与此同时也加宽了主要内容区，从而有了更多地方来放置特色图片和新闻。如图 1-3 所示。

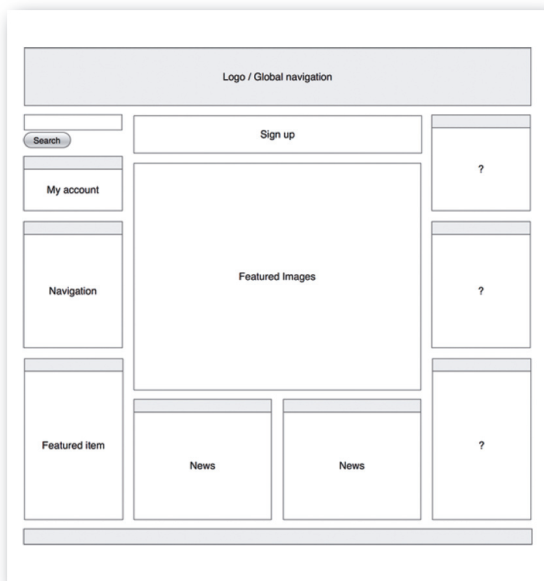

Logo / Global navigation
Search
Sign up
?
My account
Navigation
Featured Images
?
Featured item
News
News
?

图　1-2

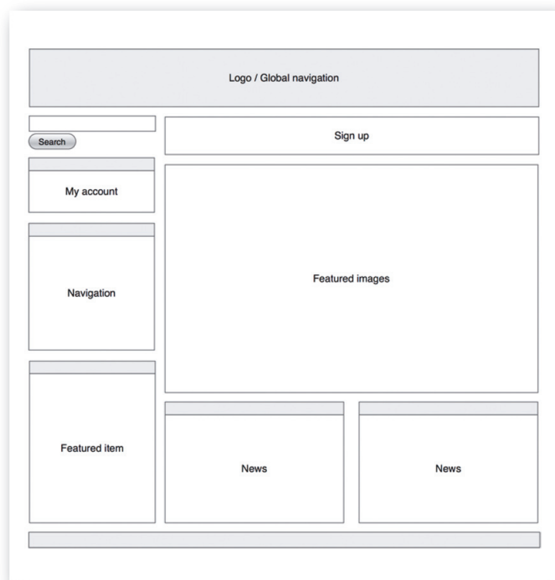

Logo / Global navigation
Search
Sign up
My account
Navigation
Featured images
Featured item
News
News

图　1-3

现在页面开始成型了。整个网页更突出了站点的主要目的——展示社区成员的艺术创作以及相关新闻。所以我确信自己的方向正确。嗯，继续前进。

我知道首页主要的目的是吸引（或者说诱使）用户注册，因此我把注意力放在了注册区域。

一般来说注册区域都需要一些颇具号召力的标语和一个按钮。所以我随便写了一些字，做了一个按钮，然后把它们丢进去。如图 1-4 所示。



图　1-4

就这么简单。

现在请允许我稍微扯一些题外话。

“Automattic”的一次飞跃

此前，我为 Automattic[①]公司主持过一次有关设计方面的研讨会。会上我们讨论了这个公司颇受欢迎的博客软件——WordPress[②]。讨论过程中我

① Automattic 公司的网址为 http://automattic.com/。——译者注
② WordPress 的网址为 http://wordpress.com/。——译者注

和 Automattic 一位叫做 Andy Skelton 的员工一起设法为 WordPress.com 首页做了一点小改动，目标是提升网站的注册率。

我们由原先的设计开始。见图 1-5。

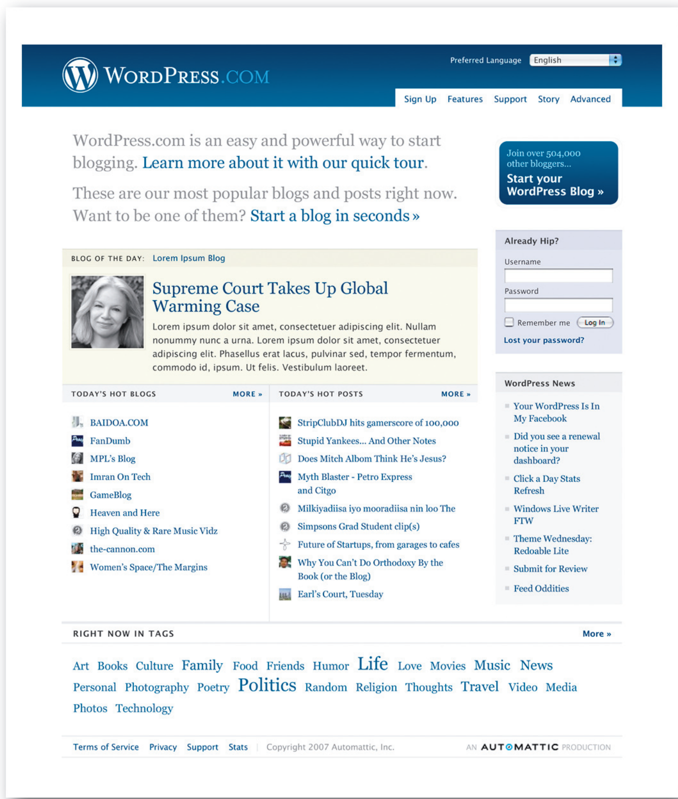


图　1-5

我为该页面创建了一个框图，便于演示和分析结构（有些地方则直接使用了屏幕截图）。关于如何让更多人关注注册区域，Andy 和我商量出了一些办法。我挪走了那个蓝色的 Start your WordPress Blog（开始你自己的 WordPress 博客），用一个很大的黄色 Banner 区域替代，并在里面放了一个夸张的 Sign Up Now!（现在就注册！）按钮。这样不仅能让用户更容易注意到注册区域，而且它把该区域和页面其他部分清晰地分隔开来。如图 1-6 所示。

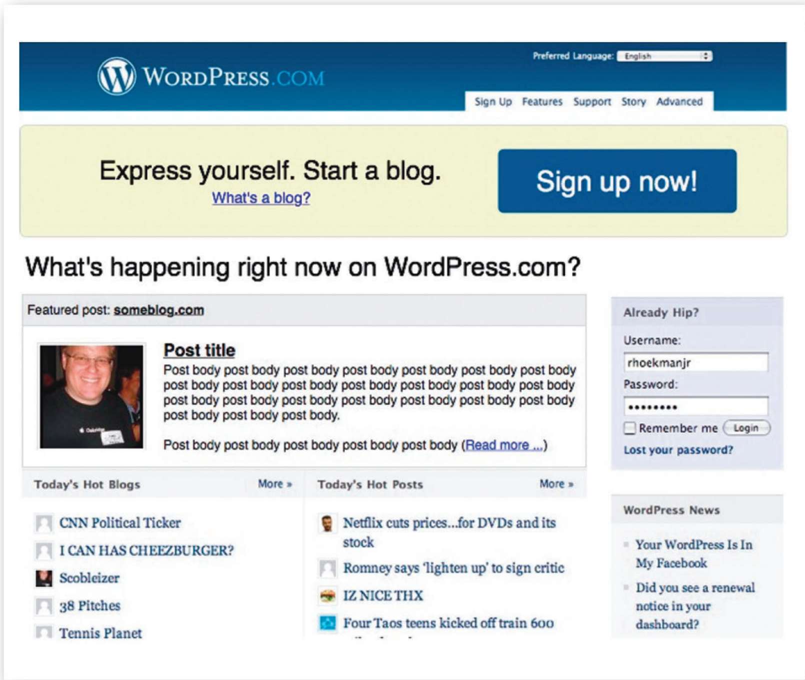


图　1-6

由此，该公司的视觉设计师兼前端开发人员 Matt Thomas 设计了新页面的最终美术效果并开始改写代码。几个小时后首页就大功告成了（这些家伙真的一点都不浪费时间！）。

设计时我们商量把 Sign Up Now 按钮改成绿色，这样它将和页面中其他所有元素形成鲜明对照，效果更加突出。Matt 试了试，感觉很不错，于是他更新了首页。图 1-7 是最终的结果。

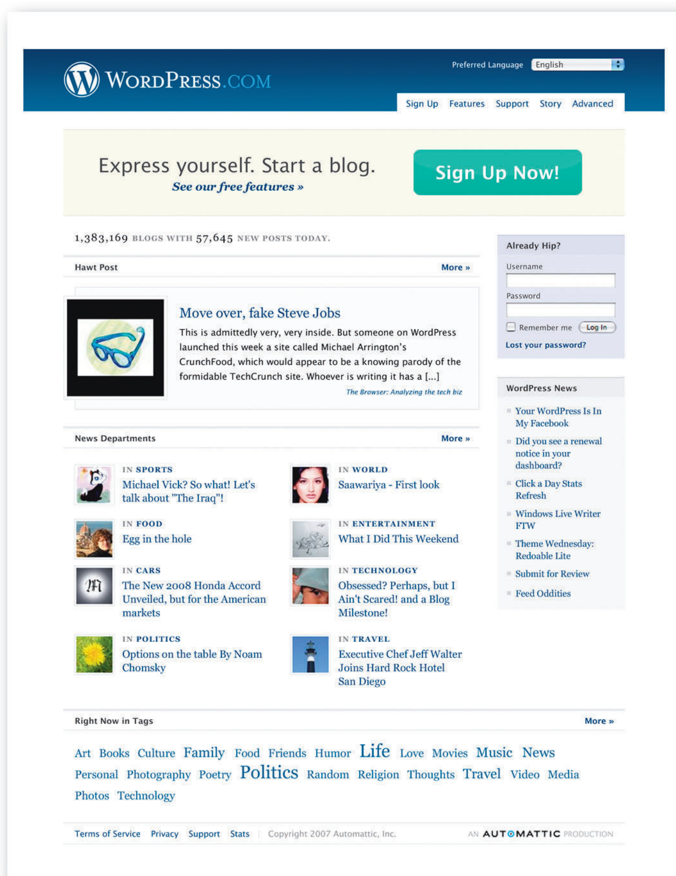


图　1-7

我们所做的工作如下：增大了注册区域，写了一句颇具号召力的标语［Express yourself. Start a blog（创建博客，表达自我）］，同时还把 Sign Up Now 按钮做得又大又美观。此外我们把这个按钮设置为绿色，这样它就能从页面中脱颖而出。这些改动创建了一条清晰的通道，带领浏览者们前往注册页面，那里也正是我们想要他们去的地方。

当时对于这个新设计能提升多少站点注册率，每个人的看法不一。而 Matt 是其中最乐观的一位，他猜测注册率将会提升 10%。

而事实证明，这点小小的设计改动产生了相当大的效果，超出了Automattic团队所有人的预期。

新页面上线后的第一天，新的博客注册率和用户注册率分别提高了大约12%和14%。这足以好好庆祝一番了，但这还不是最后的结果。

接下来的几周内，注册率提高到了25%，并且稳定地保持在这个水平。

小小的改动却带来巨大的飞跃。特别是当我们考虑到WordPress.com每个月都会迎来数百万名浏览者这一事实。所以，如果你以前怀疑过设计能够改变项目的命运，那么，现在开始相信吧。

因为这一点，我认为自己在这个社区网站上能更好地把用户的视线引导到注册区域来。

1.2 引导用户的视线

我发现，如果把注册区域移到主要图片展示区的下面，就能创造出一种被称为是Diagonal Balance（对角线平衡）的和谐状态。这个设计理念源自于“古腾堡[1]图表”。

1.2.1 应用古腾堡图表

人们在浏览页面或布局的时候，视线往往趋向于从左上角移动到右下角，就好像页面的布局设计有某种自然引力一样。因为人们通常都是从左至右、由上往下地阅读，久而久之视线自然会沿着这一路径移动。古腾堡图表简单地描述了这一阅读轨迹规律，见图1-8。

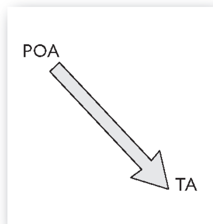


图 1-8

① 约翰·古腾堡（Johannes Gutenberg，1400—1468），德国印刷工人，传统上认为是他发明了西方的活字印刷术。——译者注

古腾堡图表中，左上角是第一视觉落点区（Primary Optical Area，POA），而右下角则是最终视觉落点区（Terminal Anchor，TA）。与之相对，右上角和左下角则是视觉盲点。因此，采用对角线平衡通常都是一个好设计所需要考虑的因素，因为设计师遵从了用户习惯性的眼动规律。

简单来说，为了实现“对角线平衡”的布局设计，我们可以运用视觉元素来创建一条虚拟的“线”，让用户的视线跟随它从左上到右下呈对角线似地移动。这条对角线就像箭头一样指向最终视觉落点。

举个例子。在 Mini 汽车的美国官网首页上，动画里各个元素出现的先后次序同样也采取了对角线设计。首先，网站的 logo 标志出现在页面的左上角，见图 1-9。

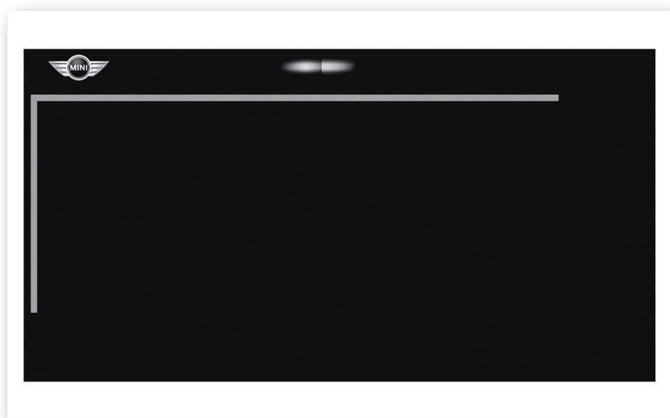


图　1-9

然后页面上画出了一个矩形线框，从左上角到右下角以动画形式出现，见图 1-10。

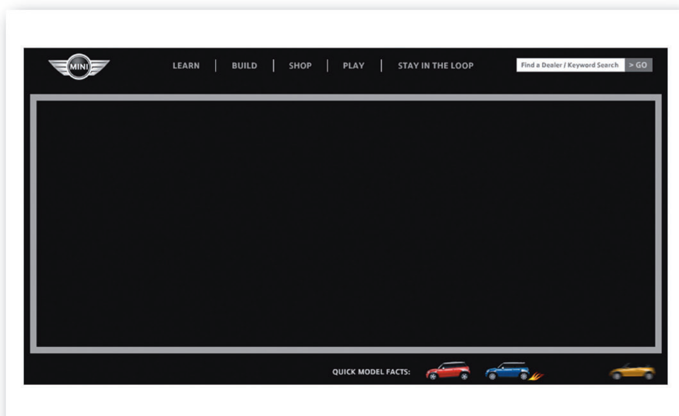


图 1-10

最后，一系列 Mini Cooper 的小车模从右下角驶入屏幕，见图 1-11。

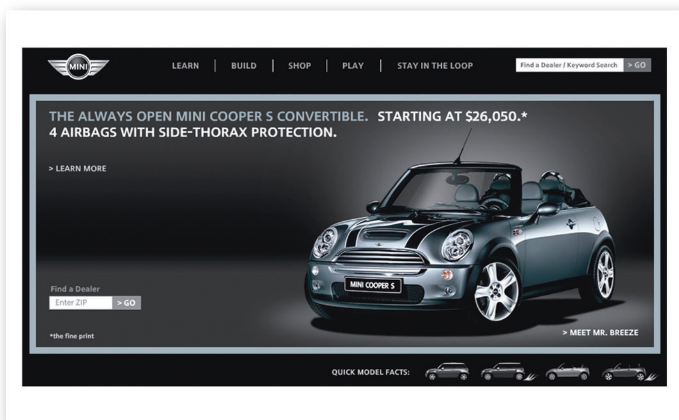


图 1-11

我们常见的对话框也同样遵循了从左至右、由上往下的视觉规律，如图 1-12 所示。

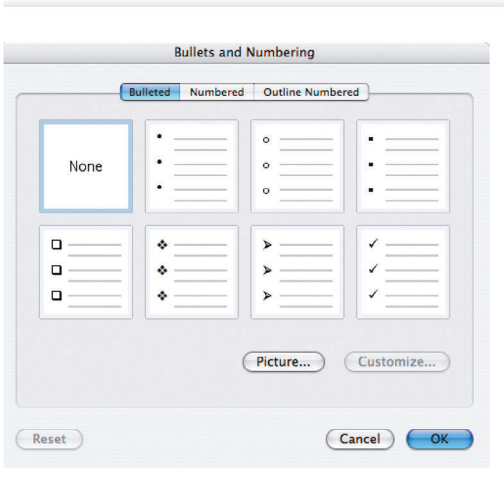


图 1-12

看上去很熟悉吧？就是这样。

为了把这一方式应用到客户的主页上，我交换了注册区和图片展示区的位置，使站点 logo 和 Sign Up Now 按钮之间形成了一条对角线，如图 1-13 所示。

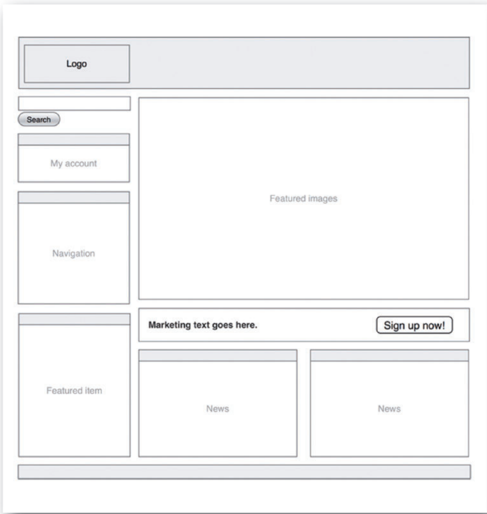


图 1-13

然而，为了让这个社区的页面真正地焕发光彩，我还需要加入一点颜色。

1.2.2 使用颜色来吸引注意

颜色能把人的注意力吸引到特定的元素上去，而且效果非常奇妙，尤其当某个元素的颜色与其他的形成反差时更是如此。相对于一致性来说，人类对物体间的差异性更为敏感。因此在页面中加入一些颜色，可以非常有效地把浏览者的视线引导到我们希望他们看到的地方。

为了着重强调在首页中采用对角线平衡设计，我把站点 logo 的主色调海蓝色应用到了 Sign Up Now 按钮上，然后把它也设置为主要图片展示区域的背景颜色。见图 1-14。

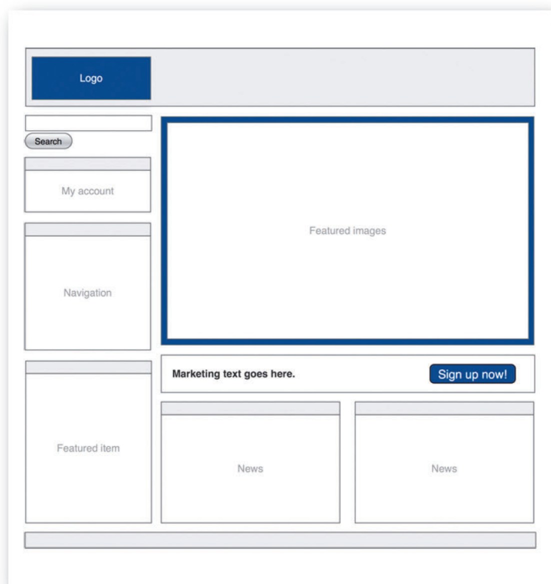


图 1-14

随后，我决定页面中的其他元素将不会再用到海蓝色。

通过使用相同颜色将三个元素捆绑到一起，用户的视线将由站点 logo 引导到充满趣味的相关内容（特色图片区），然后再移动到 Sign Up Now 按钮。

因此，从最初简单的布局设计开始（只是把需要的栏目堆放到页面上，而没有做其他考虑），我逐步地做了一些小小的变动，创建了一条对角线，从而支持了用户由左上到右下的视觉自然规律。采用视觉移动的自然轨迹意味着最终的设计将有助于用户更快地熟悉站点，并且指引用户下一步该做什么，这样会让页面更加友好。换句话说，设计让用户对网站产生了美好的第一印象。

对角线平衡法则并不一定适用于每个页面设计，因此不必强制遵守。但是如果发现使用对角线平衡可能有助于强调整个页面意图，不妨考虑一试。另外，当你在开始某个设计的时候，时刻想着对角线平衡法则，很多情况下它都会发挥效用。

常见的页面布局案例，可以参考 Yahoo! Design Pattern Library[①]的 Page Grids 栏目。那里还提供一些页面模板，在你开始设计时可能会用到（记住，关于本书里引用的所有链接，你都能在 www.rhjr.net/dtm 找到）。

① Yahoo! Design Pattern Library 的网址为 http://developer.yahoo.com/ypatterns/。

——译者注

第 2 章 凸显个性

我是一个极简主义者。过去是这样，将来一直也是这样。事实上，我曾经常幻想着有一天能够说服自己最终创造出一个简单到什么也没有的页面。

几年来，我为自己的个人网站进行了多次改版设计。而每一次改版所做的都只是在之前的基础上或多或少地简化、再简化一些。回想起来，在每一次改版完成时我都认为不可能再简洁了，可随后又推翻了这一结论。最近偶然翻出了之前某版本的备份，让我追忆良久，见图 2-1。

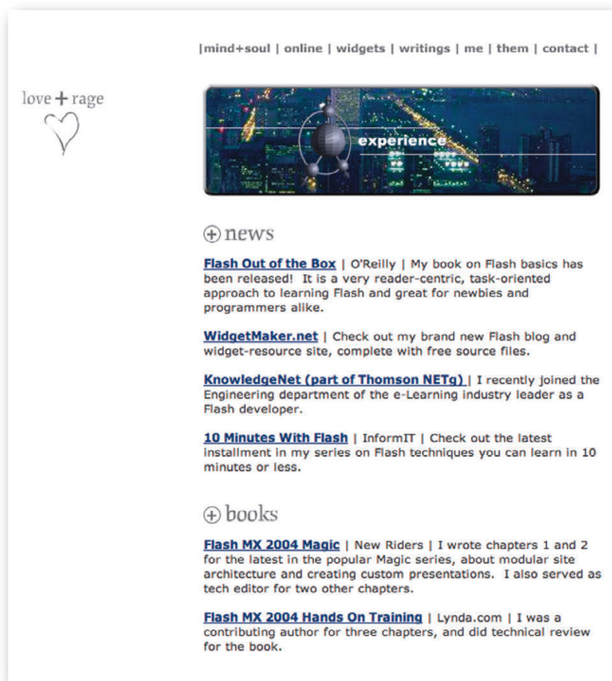


图 2-1

我的意思是说，在这几年逐步深入学习网页设计和交互设计之后，我提升了欣赏的品味，磨练了设计的技巧。而这些都迫使我去设计一个新的网站版本，使之能更好地展现我的个性。很多人都对这个网站表示赞许，但其实在一开始我特别惊讶。因为当时我并不认为它有什么特别之处——它是如此简单，如此基本，如此乏味。

当时我觉得，这个基本得几乎不可思议的布局根本就不值得大书特书。图片没几张，所以也没什么可欣赏的；然而文字却不少，所以你必须读很多。总之，在那个时候我觉得网站里到处都是缺点。

但结果证明它们不是缺点。事实上这个设计正好传达了我所希望传达的内容——我是一个极简主义者，喜欢用最简洁的方式与人们沟通（有点讽刺意味，不是吗？）。这种特别的设计风格其实展现了我的个性。

所有的网站都在展现某种个性，但这并不一定都是有意为之。很多时候个性都是在无意间流露出来的。这种时候，公司不知不觉间可能就会把内部问题暴露在网站这样一个和客户发生接触的第一平台。比如说，如果公司士气低落，网站的设计可能会表现得乏味而毫无生气；而如果雇员激情满怀兴高采烈，该公司的网站则很可能丰富多彩，活跃而富有魅力。

我一直进行网站改版其实也是一种个性的展现。每个网站都可以这样做来展现自己的品牌形象，或者说传达一个公司希望和外界交流的讯息。

设计时我并没有真正意识到我正在干什么，我只是在不停地摸索直到确定某个想法。但是这种极端简洁的设计最终使我发现，运用一些巧妙的元素把所有设计捏合成一个整体将有助于展现个性，不管团体或个人都是如此。而且很容易做到这一点：我们只需要强调那些已经有的东西，从而形成一种有凝聚力的“品牌”。

2.1　统一设计风格，形成积极的印象

几年前刚开始设计时，我知道页面应该采取一个单列的布局。因为我要放在网页中的文本只有一列，同时站点导航部分也只需要一行。因此我创建了一个简单的居中列，它的宽度足以在低分辨率的显示器上显示，而且左右两边还能留下很不错的边距。如图 2-2 所示。

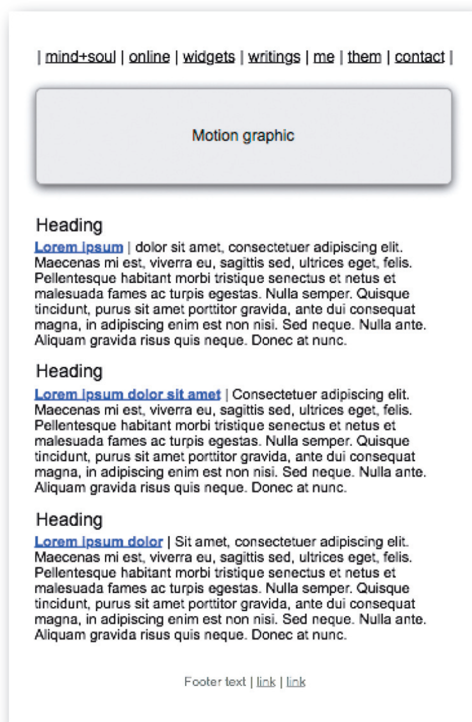


图　2-2

首页顶行供站点导航使用，而它下方的空间则预留放置动画图片（那个时候，我做过很多动画设计）。

由于在文本列的两侧都留下了很宽的边距，我决定在左边区域放置站点标志。标志是一个非常简单的 logo，包含了两个单词 Love（爱）和 Rage（怒），由一个加号（+）隔开。这是当时我从巴兹 · 鲁霍曼执导的莎士比

亚名剧 *Romeo + Juliet*①电影海报中获得的灵感。是的，我知道从一部视觉如此丰富的电影中获得灵感非常容易。但是朋友们，只要能有点灵感我就已经很快活了。

在这个 logo 下面有一个小小的、手绘的心形图标，从而构成了整个标志。顺便说一句，“爱与怒”这个点子源自于我多年前写的一首歌（显然，它也是一部电影的名字②）。再一次，我因为灵感而兴奋，即使那是出自于我自己写的歌词。此后我在网上再没有使用过这个短语。

我把这个图形放到结构框图里，把它与动画图片顶部对齐。见图 2-3。

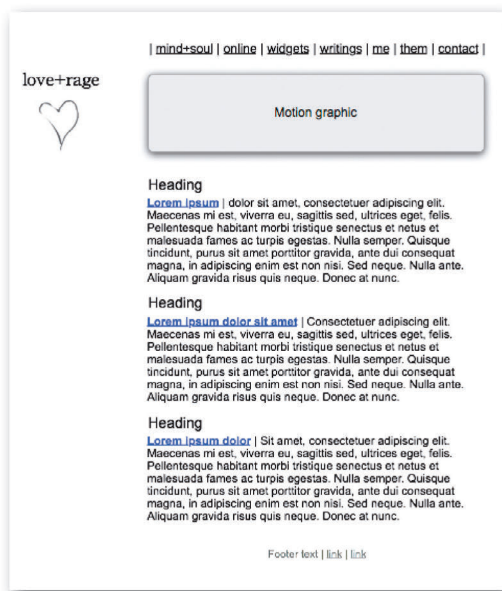


图　2-3

这是个好的开始，但它还不足以把所有设计捏合成一个整体。我希望有什么东西能够在整个站点中一直延续下去，并且整合每个页面，为网站创造出一个有凝聚力的形象。我所想要的应该是一个符号。

---

① *Romeo+Juliet* 引进版名为“现代罗密欧与朱丽叶”，美国福克斯影业公司 1996 年出品。在电影标题中，“罗密欧”和“朱丽叶”之间有一个“+”的符号，里面有一个小小的“&”。相关网址为 http://www.romeoandjuliet.com/。——译者注

② *Love and Rage* 是一部德国与爱尔兰合拍的影片，Nova Films 于 1998 年出品。——译者注

使用符号

我第一次了解符号语言是在看电影的时候。不，不是那些角色[①]，而是角色之外的东西。它们就如同贯穿影片的隐喻和象征一般用于表现某个主题。

例如，在影片 *In the Cut*[②]（Pathe Productions 于 2003 年出品）中从始至终都是黯淡的色调，但偶尔会插入一抹深红来勾勒出主角 Frannie（由 Meg Ryan 饰演）的线索。影片的标题、故事情节和结尾的那一幕都无可避免地与红色联系到一起。本质上颜色成为了影片的一种符号语言，或者说是另一个角色。

同时，我从自己的偶像设计师 Hillman Curtis 身上学到了更多关于符号的使用方法。

当 Hillman 和他的团队为影片 *Bend It Like Beckham*[③]（Gurinder Chadha 于 2002 年执导）设计官方网站时，把网站的焦点集中在一个足球上，让它带领着用户从网站的一个栏目移动到另一个栏目。这个足球符号整合了所有的设计元素，同时延续了影片中那种乐观向上的情绪。见图 2-4。



图 2-4

① 此处“符号”的原文为“character”。英文中“character”含义丰富，除了“符号”之外还有“角色”、“字符”、“性格”等多种字义。——译者注

② *In the Cut* 中文译名为“裸体切割”或“凶线第六感”。相关网址为 http://www.apple.com/ trailers/sony_pictures/ in_the_cut/。——译者注

③ *Bend It Like Beckham* 中文译名为“我爱贝克汉姆”。Kintop Pictures 于 2002 年出品。网址为 http://www.foxsearchlight.com/benditlikebeckham/。——译者注

然而，并不是所有符号都能像摄影师运用视觉叙述那样动感和多姿多彩。在大多数情况下，他们只需要简单地在网站中重复出现，以某种延续的趋势来保持一致的主题风格。

在这个项目中，我有一个加号图案和一颗心。我最先尝试的是心，把它放在页面每个标题的前面，就像段落的着重号一样。见图 2-5。

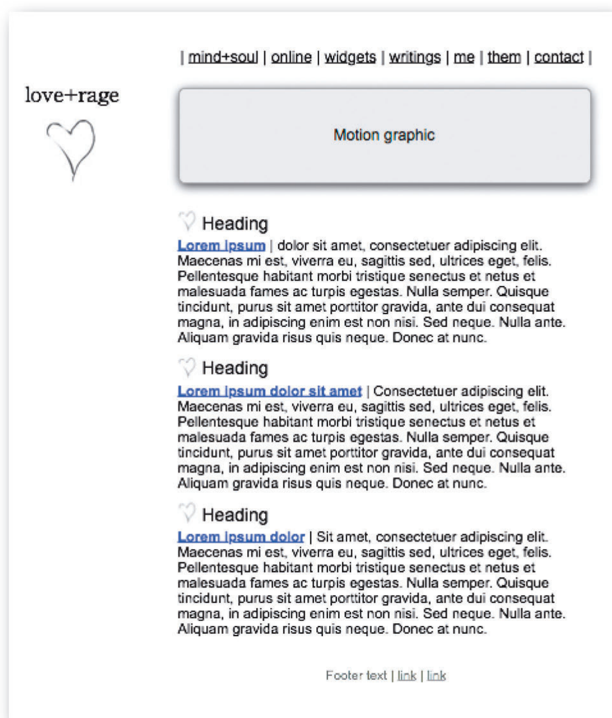


图　2-5

老实说，这样看上去相当愚蠢。自从初中不再帮人给女生递纸条后，我就没见过在哪个地方有这么多心的。这可不是我想在网上用来表现自己的那个图形。于是我抹去了这些心，试着用经过加工的加号图案来替换它们。见图 2-6。

加号图案有两层含义。它不仅仅是网站 logo 元素在整个站点中的延续，还创造出一种效果，即页面中的每个文本段落都吸附到各标题的下面。与此同时，它还暗示这些内容是新的，仿佛那个“+”意味着“刚加上”一样。

很好，这个小成果让我感到非常幸运。运气绝对是设计的一部分。在设计 logo 时我根本没打算重复使用这个元素，但当它出现在合适的位置时我发现效果很棒，何乐而不为？

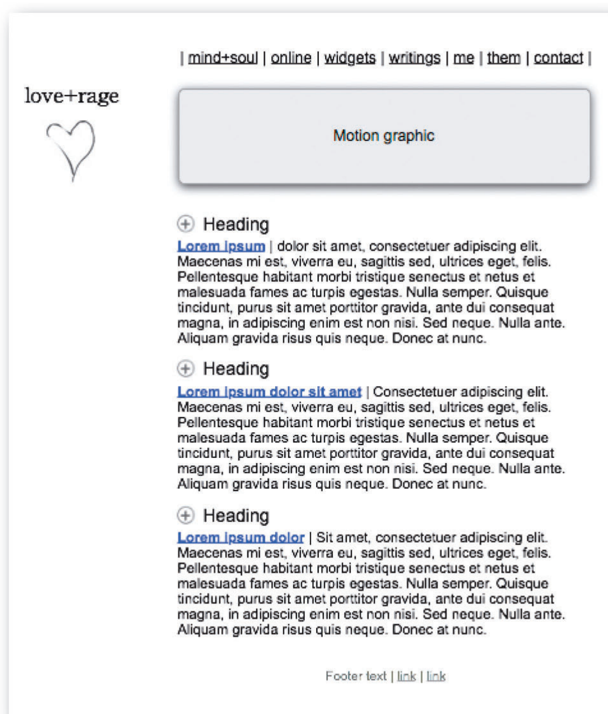


图　2-6

同时我还注意到页面中最突出的 logo 和标题文字，它们之间有点不太相配。这个矛盾让我意识到还得再深入一步。见图 2-7。

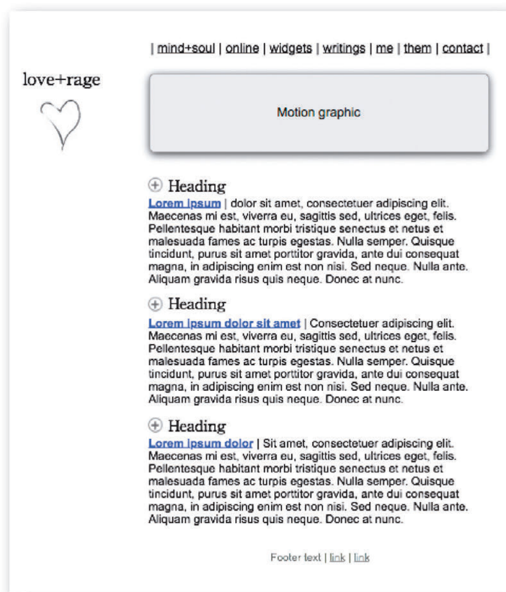


图　2-7

我需要在标题中使用和 logo 一样的特殊字体。于是我删除了各标题的 HTML 文本，而把文字做成图片显示。在那个时候我并不太了解易用性和搜索引擎优化技术，也没有意识到所用的都是可恶的表格[①]。匹配字体并不算错——但是把文字转化为图片意味着搜索引擎蜘蛛无法对标题进行分析，而且这些图片对屏幕阅读器（帮助那些视力不佳的用户浏览网页的设备）[②]来说也毫无意义。

当时我的确很天真，就这么艰难地行进下去。我编写了页面模板，创建了动画图片，然后为站点构建了所有其他的页面。

① 当今流行的 XHTML+CSS 技术要求 Web 设计师使用规范的 HTML 代码。W3C 组织建议 HTML 代码应当只起到单纯的语义作用，这表示页面布局应通过 CSS 来实现。严格意义上来说，表格只能用于表现表格数据，而不能被用来规划页面的布局结构。这对广大习惯于使用表格布局的设计师来说是一个很大的冲击，但亦是潮流趋势的必然。——译者注

② 屏幕阅读器的原理是“朗读”出页面中的文本文字。很明显它们无法朗读出图片的内容，但可以读出图片的 ALT 字符串。（HTML 中为 IMG 标记添加的文本字符串，用于对网页中的图片进行描述。它允许使用纯文本浏览器或选择不下载图像的用户了解图像的说明。）——译者注

为了延续加号图标这一符号，我把它用在每一个页面的每一个标题上。此外，与站点名爱与怒（love and rage）相符，另外一个栏目的主题 mind and soul 中也有一个 and，我把这个图标也用在了那里。见图 2-8 的左侧。

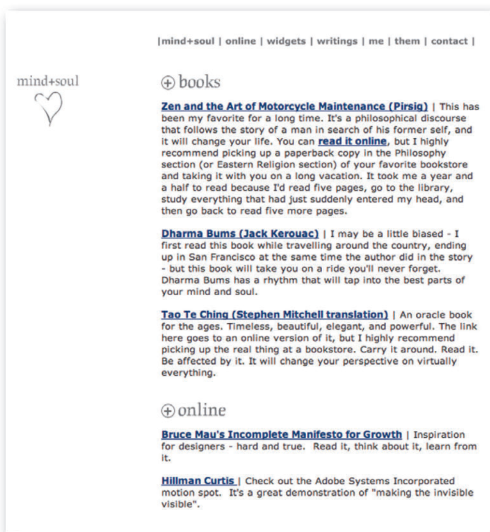


图 2-8

就是这样了。这个简单、基础、乏味的网站，满是缺陷和失误。但它仍是我的最爱，胜过任何其他我为自己设计的站点。

这段时间以来，我一直希望重新设计自己的网站，能够像这个最初的设计一样展现我的个性。而下一次设计，我希望自己能更慎重、更周到，让新的版本成为我的最爱。

就像我所说的，不管是有心还是无意，网站都会展现某种个性。时刻注意网站正在传达什么，并相应做出简单的改动，我们便能够往自己想要的任何方向修塑我们网上的形象。一般来说，大部分人都不愿自己看上去单调乏味、毫无生气，每个人都希望传达给用户一种特别的氛围。品牌标志以及源自品牌标志的符号，能够以一种文雅而易于理解的方式来传达信息，并且向用户展现我们真正希望他们看到的东西。

也就是说，让你的品牌标志保持精致和优雅。你不能随便扔出个足球然后就指望它腾空而起不再落地。我们需要有良好的判断力来调整品牌商标和符号——不要太突兀，有时候宁可让它们简单朴素一些。

第 3 章
导航之道

在你第一次使用某个 Web 应用产品时，熟悉它的关键在于是否能理解整个站点的内部结构。而最初的几个瞬间对于是否能形成美好的第一印象至关重要。

令人迷惑的术语和拙劣的结构组织，将会在几秒钟内把我们从满怀希望和兴致勃勃带往灰心、失败和挫折。所以能很快在网站中找到方向的重要性是毋庸置疑的。

人们通常获得导向的方法是：粗略扫视一遍页面，看看自己有哪些选择，并且快速过滤掉那些无关自己兴趣的内容。例如，在某个企业网站寻找该公司的信息时，我们会寻找诸如 About（关于）、Company（公司）、Us（我们）以及类似的超链接。如果看到那些不符合自己需要的链接，我们会直接忽略它们。而当某个符合需要的链接出现时，我们会更为仔细地查看与其毗邻的文字，以图确定其是否正是我们需要的选项。如果是，我们就单击。

换句话说，我们先浏览页面的各主要区域，找出相关的部分，然后在这些部分里的项目中找出关联性更强的，并且希望最终能发现那些我们确实一直在寻找的内容（或者，如果浏览没有什么特定目的，我们可能会发现更有趣的内容）。

这种推理的过程不仅能帮助我们首次浏览时找到方向，而且还能在我们使用过网站一段时间后，帮助我们进一步寻找某些信息，或者完成某些任务。出于这种原因，导航过程本身是否合理变得至关重要。它的意义不在于是否能反映了网站内部的结构，而是在于对我们用户是否合情合理。网站必须符合我们思考问题的方式。

我曾经参与过一个项目，需要重新设计一个面向网站建设者的“所见即

所得[①]”式的 Web 应用。这个程序有些类似 Google Page Creator[②]，它让用户仅需操作可视化的编辑工具就能创建和编辑网页，而无需手写代码。

然而，与 Page Creator 不同，这个产品已经存在了很长一段时间，经历过好几次大的改动。站点的功能越来越多，站点的导航也变得越来越繁冗。网站建设者只是随意地将新的界面元素添加到网页中，结果用户越来越难完成任务。

对于一个用户达好几万人的 Web 应用程序来说，这种状态可不乐观。

开发团队和我在一起工作了好几周，对该产品做了一次全面检修。导航功能只是其中的一小部分，但却又是非常重要的一部分。

3.1 告诉软件该做什么

在我准备着手设计整个产品的导航（或者称之为持久导航）时，它看起来其实相当简单：只包含两个按钮标签，分别是 Home（首页）和 Site（站点）。然而展开 Site 标签后，却出现了一大堆应用功能选项，而且菜单项之间毫无顺序。见图 3-1。

真正的下拉菜单要比这里显示的长很多，但这已经足够让你发现问题了。每一个菜单项都为了一个完全不同的目的而通往一项完全不同的任务。例如，Create new page（创建新页面）和 Edit contact information（编辑联系信息）选项，很明显相互之间毫无关系。没有一个用户会这么想：我要编辑联系信息，所以应该到 Site 菜单里面找。但事实上这两个选项居然就像两只走丢的小狗一样紧紧靠在一起。

这些不同类型的动作应该以更自然的形式进行分组。从这个菜单往下看，我发现了一些彼此相关的名词。诸如 Web page（网页）、New page（新页

---

①“所见即所得”的英文缩写为 WYSIWYG（What You See Is What You Get，通常发音为“wizzywig”）。它是一种计算机方面的技术，使用户在编辑视图中所看到的文档与该文档的最终产品具有相同的样式，也允许用户在视图中直接编辑文本、图形或文档中的其他元素，故也称“可视化操作”。例如 Dreamweaver 等软件。——译者注

② Google Page Creator 是 Google 公司于 2006 年 2 月推出测试的在线网页创建工具。人们可以在其免费提供的 100MB 空间中自由创建网页并托管。该产品提供多种网页模板，用户可以直接输入内容、上传图片并发布这些网页，无需懂得 HTML。网址为 http://pages.google.com/。——译者注

面）和 Navigation（导航栏）等都是可以创建和编辑的站点组成部分，而 Templates and colors（模板和颜色）以及 Fonts（字体）则与网站的外观有关联。

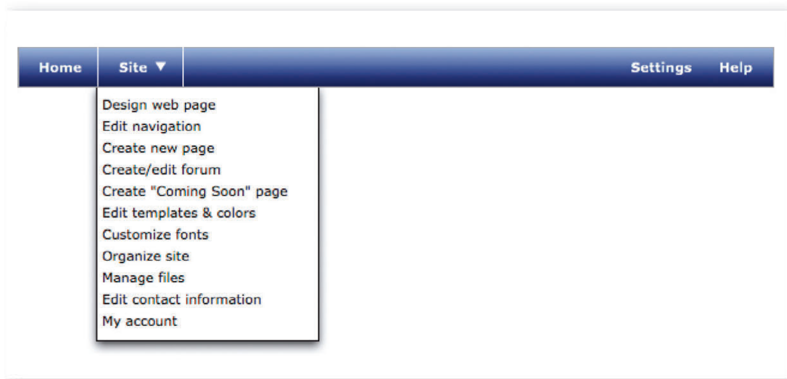


图 3-1

于是我把这些菜单选项分成了几个部分，使之更好地反映了它们彼此间的自然联系，然后对它们分类标识。新的栏目标签如图 3-2 所示。

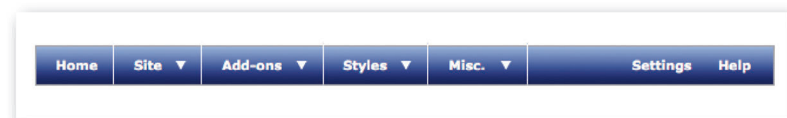


图 3-2

这些新的分类标签把一开始的选项列表划分成好几个组，分别包含了站点各元素、扩展功能（例如论坛），以及网站样式。然后，重新检查，看看每个菜单在新的标签下表现如何。在做这些的时候，我还去掉了每个菜单项前面的动词，比如 Edit（编辑）、Manage（管理），等等，只留下了简单的词组，这样它们可以更容易地被用户扫描到。用户只需要浏览图 3-3 中的列表，就可以很轻易地找出 Web page、Navigation，或者任何其他选项。

分组看上去变得更有意义了。然而，就像你所看到的，我同时也创建了一个怪里怪气的“废料箱”菜单，我称之为 Misc.（杂项）[①]。

① Misc. 为 Miscellaneous 的缩写。——译者注

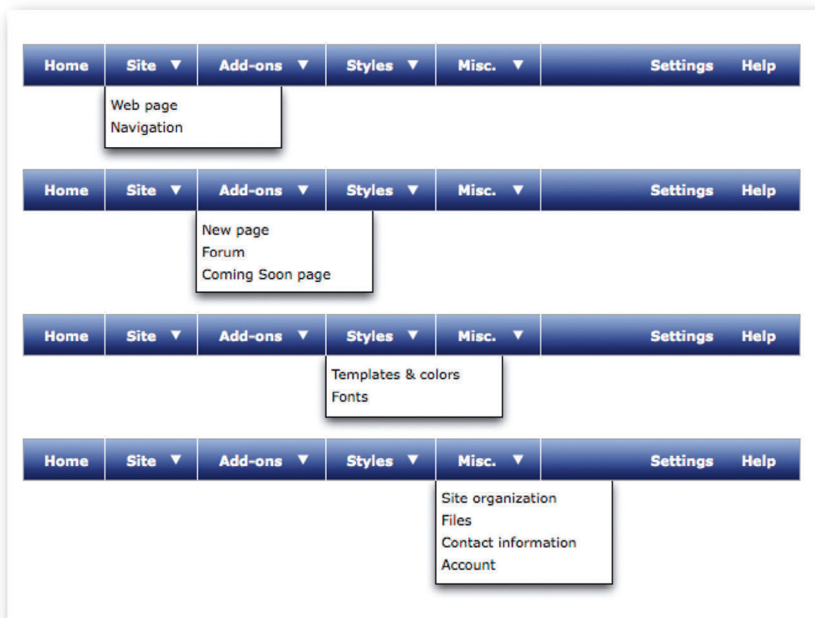


图　3-3

每个应用程序都需要一个废料箱，对吧？

是的，我也不喜欢它。在头 5 秒钟内它还貌似有点意义，但随即我就意识到这玩意很令人厌恶。首先，我很想找一个更好的名词去表现 Misc. 里面的项目，而随后我发现是否有好名词这一点根本不是问题所在。

3.1.1　每个人都想当指挥

软件并不是为了我们能单击那些按钮而存在。它是为了我们能完成任务而存在。这样我们才能做事情。

所以导航栏不应该是陈列在一排按钮上的一串名词，而应该帮助我们感觉自己正在做一些事情，让我们找到当司机的感觉。

我们人类不喜欢感觉到自己正按软件的要求去做事情。在决定采取某个行为的瞬间，我们希望由自己控制软件，由自己告诉它应该做些什么，而不是相反。任何违反这一意愿的软件都会让人感到愚蠢而糟糕，结果会敬而远之。

看看你的身边，你会发现很多软件的导航都设计成让用户感觉自己才是主人。从微软的 Word 到 Adobe 的 Illustrator，它们的菜单所包含的选项都是为了用户去做某件事情，例如 Edit（编辑）和 View（查看）。

这些菜单不是基于对象而是基于任务的。这才是它们应该有的样子。

我关注并重组了菜单中的对象，但其实我应该关注这个程序中所需要完成的任务类型。为了做到这一点，那些术语应当显示为**动宾词组**，而不仅仅只是又丑又过时的名词。

也许你已经明了，所谓动宾词组就是动词 – 名词的简单组合，以动词开头并以名词结尾。通过这种形式构成的词语，我们就能告诉软件我们希望做哪种动作，以及这个动作所针对的对象是什么。这让我们感觉自己正在指挥软件：我们是领导者，而不是被领导者。

如果没有动词，菜单就很愚蠢而糟糕。

为了补救这一问题，我把由名词主导的按钮标签改成了由动词主导的标签。见图 3-4。

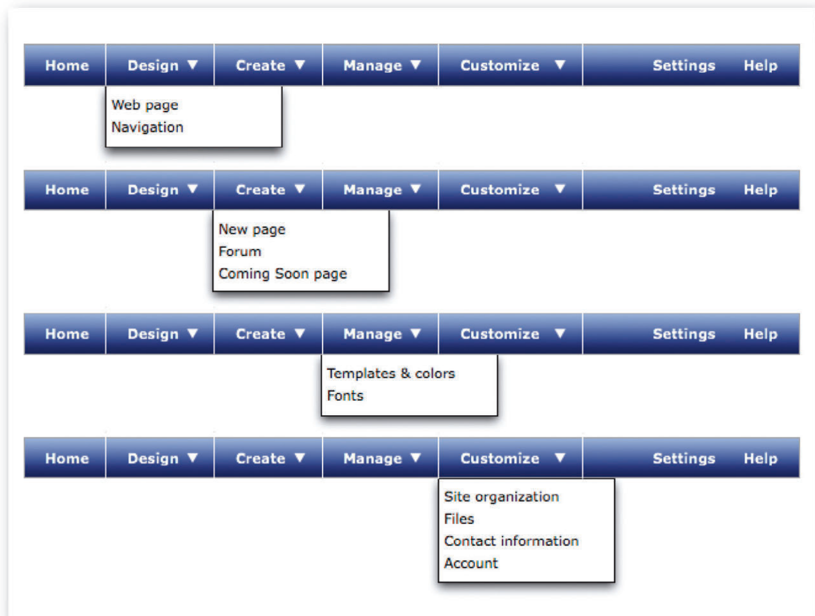


图 3-4

不再是为了任务而去选择 Styles（样式）→ Fonts（字体），你现在可以告诉软件你想 Manage Fonts（管理字体）。按钮标签是动词，而菜单选项是名词。

现在用户可以感觉到自己在指挥着软件。现在他们能够去 Design（设计）导航栏或者某个存在的网页、Create（创建）论坛或新页面，以及 Customize（定制）站点的结构、文件等。

顺便说一句，如果用户希望为网页中的文本选择某个字体，我也许会建议使用 Styles → Fonts。但在这个项目中，用户希望的却是管理各种可用字体，以便于能转化为图片显示，而不受其他浏览者系统字体的限制。

在把这一块转交给开发团队之前，我还需要再做一件事。

3.1.2 避免 Login 综合症

每个设计师或多或少都会有自己的怪癖。我的怪癖就是十分讲究怎样使用 Login 以及与之对应的 Logout。

我参与过许多应用程序的开发，它们大多都会用到这些词儿，但其效果总让我怀疑。

所有策划、设计和构建 Web 应用程序的人都想要实现一些目标。要实现起来并不轻松，必须怀着创造一款伟大的产品的信念，而且要全心投入其中才行。

开发者、项目经理还有设计师们创建一款产品，花费了无数个小时去规划、编写代码、测试可用性，还要考虑到所有能想到的事情。而在这些难以想象的艰苦工作之后，他们把所有的内容交给一个碰上谁就是谁的人来发布。而这个人往往会输入一个 Login，然后就这么扔在那里。看看图 3-5 吧。

其中两张截图来自非常知名的 Web 应用产品。在这些首页上就有一个很基本的语法错误，而网站的每一个用户都会时不时看到它并且会单击它。也许是每周用到一次，也许是每天就用到一次。

在这些截图里面，还有互联网上数不尽的其他类似例子中，Login 这个词的用法是错误的。

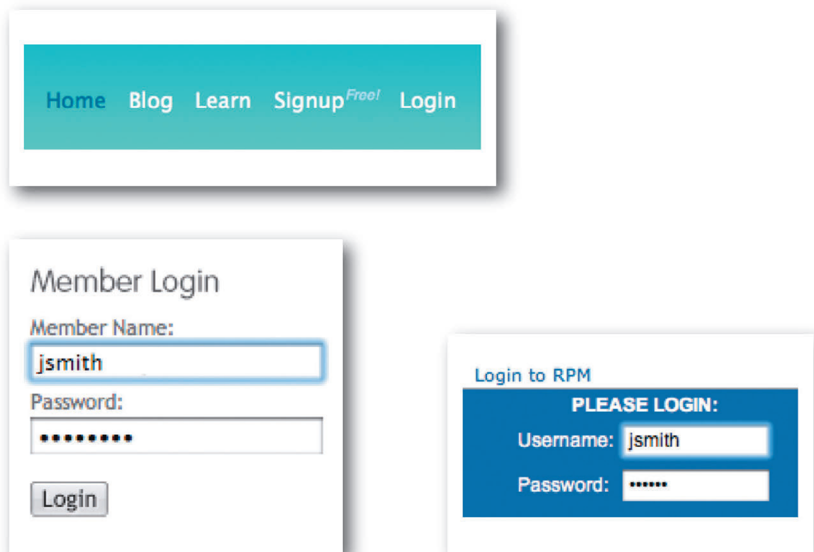


图　3-5

Login 不是一个动作，这就好像“鞋”不是一个动作一样。它是一个名词，意思是 the Login Interface（登录的界面）。

你不会想要用户去 Login，那样根本毫无道理。你想要的是让他们去 Log In。在产品界面中使用 Login 一词表现得很不专业，说明开发者没有尽到责任——在程序中有一些内容没有经过仔细考虑，没有确保所有的设计都是精心而为的。

这就像竖起一块牌坊告诉大家：“我们很细心，但不是百分之百。”

再加上，用户所希望的并不是去和界面里的对象进行交互，而是去做自己所要做的事。他们的目的并不是单击它然后去看登录面板，而只是想登录。

因此你所需要的不是一个名词，而是一个命令。

我的这个项目里面也有 Login 这个词，见图 3-6。

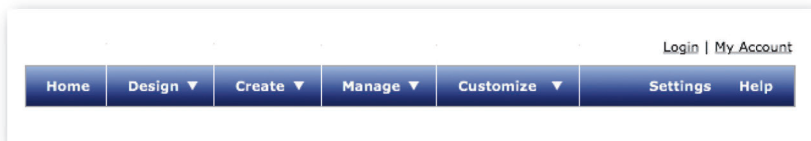


图 3-6

所以我把它变成了一个命令，见图 3-7。

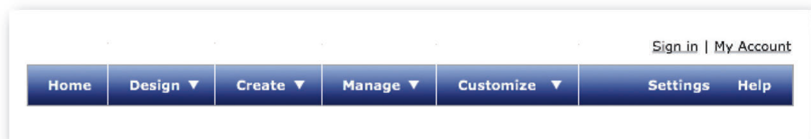


图 3-7

也许大多数人从来都不会注意到这里所显示的是一个单词还是两个。但即使没有人注意，错误的用词仍然会导致产品给人留下不专业的印象。差之毫厘，谬以千里。请确保产品的每一个细节都是正确的，以此来告诉用户：

“我很细心，百分之百，从始至终。”

3.1.3 我来说，你来做

对于用户来说，他们对站点保持兴趣的关键在于能否获得方向感、能否得到所需的信息，以及能否完成任务。清晰的导航结构不仅有助于用户了解网站能做些什么，还能知道如何去做。

运用动宾词组，按钮和链接就能有效地告诉用户能做什么。同时也使得用户感到自己是指挥者：是用户告诉应用程序去做什么，而不是反过来。

创建产品导航时，请确保你是以用户为出发点，而不是以系统为出发点。用户不是工具，系统才是。

第 4 章
链接生来不平等

无论你习惯于在哪个网站浏览新闻，比如 BBC 新闻网[①]、CNN[②]、Yahoo[③]，或者其他什么网站，你都会面对各种各样的链接。这些链接会按照时间、相关度、点击率以及其他种种顺序排列。这是件好事，因为它使得找寻信息相对来说更为容易。见图 4-1。

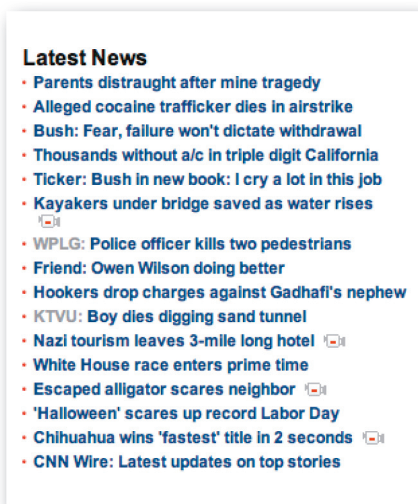


图 4-1

这说明即使是链接这种最简单、最基础的设计也是能够改进的。最近，在创建一个链接列表时，我找到了一个这样做的机会。

① 网址为 http://news.bbc.co.uk/。——译者注

② 网址为 http://www.cnn.com/。——译者注

③ 网址为 http:// www.yahoo.com/。——译者注

在这个例子中，我需要在一家公司的内部网首页上创建一个标题列表，让该公司呼叫中心的雇员们随时了解最新事件。这些标题的链接有点类似新闻列表，需要按时间顺序显示，但一次只需显示 5 条信息。

我从最普通的设计开始，创建了一个简单的链接列表，如图 4-2 所示。



图　4-2

下一步考虑如何让用户看明白这些从上到下的标题链接是按照什么顺序排列的。在这个呼叫中心里，最新事件往往是最重要的事件。

我可以在旁边放一些说明性文字来做到这一点，比如“本列表按照从最新到最老的顺序排列”。但是这样需要花费用户更多的精力去阅读。对于 Web 应用来说，不必要的阅读是人们最不愿意去做的事情。

所以我采用了另一种方案。我把标题从 1 ～ 5 进行了标号，觉得这样差不多算是大功告成了，如图 4-3 所示。



图　4-3

对列表项目进行编号基本上能表现出它们排列的顺序，但是列表看上去仍然很单调。这让我相当头疼，因为即使每一个项目都编了号，它们彼此之间还是太相像了。

在这个呼叫中心里，最新事件往往最为重要。它通常意味着公司某个环节发生了突发性事件，需要尽快通知客户或消费者。这些事件可能是售后服务资源短缺，也可能是计费系统暂时性故障，也可能是其他事情，它们都需要呼叫中心的雇员在第一时间内获得上述信息。

而在这样一个单调的列表里，每一个链接都和其他链接如此相似，它怎样才能凸显出那个 No.1 的重要性呢？

最近在 Boxes and Arrows[①]上读到了一篇名为“环境的提示”[②]的文章，作者是 Ross Howard，我从中获得了些许灵感。

4.1　让环境提示成为引导的线索

Howard 在文章中提到，在日本的城际铁路线上，每一个站台都会在列车进站后播放自己独特的音乐旋律。因为那些经常乘坐同一线路的乘客（例如上班族）会在乘车途中感到无聊，他们会看书、读报纸，或者打开笔记本电脑工作，而不一定会去关注每一站的站牌或列车的语音播报。而到达每一站都提前播放一段独有的音乐旋律，能有效地引起乘客的注意，从而提早准备下车。

Howard 把这种音乐广播称之为**环境的提示**。列车线路中还包括其他的环境特征，例如隧道、每个站台的外观等。而这些音乐旋律和它们一起帮助乘客在脑海里构建一张虚拟的乘车线路图，从而避免坐过站。

用环境作提示这个想法引起了我的兴趣，因为它表示当用户在网站上完成某项任务时，很可能是一种无意识的举动。在这个企业内部网上，最有帮助的事情莫过于让用户很容易地找到正确信息。于是我尝试把环境提示这一概念应用到链接列表中。

我舍弃了列表编号，同时根据文章发布的时间顺序改变了每一个链接的字体大小。最新发布的链接字体最大，下面的链接字体依次变小，以此表现列表中的文章越来越陈旧，其重要程度也越来越低。见图 4-4。

---

① Boxes and Arrows 是一个网站，关注图形设计、交互设计及信息构架的各种实践、创新和讨论。网址为 http://www.boxesandarrows.com/。——译者注

② 这篇文章全名为“Ambient Signifiers: How I Learned to Stop Getting Lost and Love Tokyo Rail”，于 2006 年 9 月发表。原文见 http://www.boxesandarrows.com/view/ambient_signifi。——译者注

这样一来，事情发生了很大变化。较大的字从较小的文字中凸显出来，使得列表中第一个链接的重要性一目了然。另一方面，更大的字体显得更为沉重，颜色也更深，这使得它们的表现力更为突出。



图　4-4

展现岁月的印记

设计的时候，我突然想起设计师 Brendan Dawes 曾经在他的 *Analog In, Digital Out*[①]一书中谈到过岁月感的问题。Dawes 多愁善感地回忆起在他刚接触电脑时曾经读过的一本小册子，那上面不小心留下了一些喝茶时溅落的污迹。他以此谈到在数字领域中岁月感的匮乏。数字化的文件不会溅上茶污，这一点相对于实体化的真实文件来说是一个巨大的优势，即它们在创建多年后依然保存如新。但这同时也意味着从满是茶污的物品上获得的怀旧之情已经永远遗失在数字世界中了。

当时这个想法让我浮想联翩。这很自然，因为——哇喔！想想你能在网页上变出怎样的戏法！

网页上无法表现出岁月感这个问题确实一直都在困扰着我，但我从没想过那些古老的东西需要以某种方式让它们看起来古老。没错，许多网页的文章都会显示日期来表明它们的原始发布时间，但是仅此一个表明年代的迹象通常很轻易就会被忽略，而且还有很多网页根本就不显示日期——没有任何表明年代之处。

比如说，我三年前写的一篇文章，没准在今天就会变成某个搜索条件下的第一条结果。而如果文章中没有标明发布日期的话，又怎样才能让人

① 尽管书名中包含了“模拟输入”和“数字输出”的文字，*Analog In, Digital Out* 其实是一本关于如何提升交互设计创造力的书。New Riders 出版社于 2006 年 9 月出版。——译者注

们搞清楚它并不是昨天才发布的呢？

因此，我不时会收到几封电子邮件问我一些三年来从没再想过的问题。这些问题非常新，然而我以前的回答……可能就会显得有些迟钝了。

如果网页中表现出岁月的印记，那么读到这篇三年前文章的用户就能很清楚地知道它是很久以前发表的，于是就会转去其他地方搜索更新一些的信息，或者至少调整一下要发给我的问题。

在参考了 Brendan 的想法之后，我决定让这个链接列表的设计更进一步，努力传达出每个链接是更新还是更旧。毕竟在这个项目中，越新意味着越重要。

但如何能让链接看起来很旧呢？

Google Reader[①]对此提供了一种答案。它正好在当时发布了一个新版本，而我注意到在里面有一种特别的标签云[②]，它比大多数的标签云在内涵上要更丰富一些（有关标签云我们将在第 5 章专门讨论）。除了字体大小之外，这种标签云还使用了颜色来表明某个链接相对的受欢迎程度。见图 4-5。

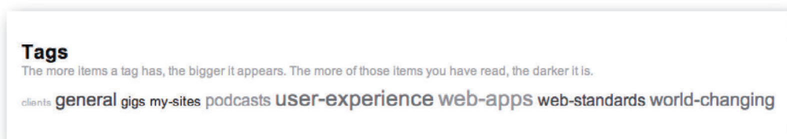


图 4-5

在它下面有一行说明文字："标签含有的项目越多，该标签字体就越大；项目点击的次数越多，该标签颜色就越深。"

我觉得这段说明文字稍嫌啰嗦，但同时我也觉得这种双重指示器（字体大小和文本颜色）是一个很好的想法，所以我借鉴了它。

① Google Reader 是 Google 提供的在线 RSS 阅读工具。用户可以自己创建阅读名单，及时获得自己挑选的内容或更新的信息聚合，也能通过电子邮件将这些聚合的信息与朋友即时分享。网址为 http://www.google.com/reader。——译者注

② 标签云的英文为 Tag Cloud。标签（Tag）是个人所做的关键词标明和归类，便于对互联网上的内容进行搜集和整理。标签云则是所有标签的集合，通过标签的字体大小来表现其关联的文章多少。它被称为是又一个 Web 2.0 元素。——译者注

我根据从新到旧的顺序修改链接列表，使得它的颜色由深至浅，如图 4-6 所示。



图 4-6

最终的调整创造出一种效果，即从最新的链接开始逐渐向下淡化，而这也使得表意更为清晰：第一条链接最重要，从它开始，链接的重要程度逐条递减。

非常好。这个正是我一直想要的。

这种环境提示在最终的网页中是不是真的起到了明显的作用？或许并没有。会不会有人因为列表最后的链接字体太小而烦恼？有可能，但是因为浏览器字体的大小可调，所以这种情况也不太会发生。我是不是在劝说你应当立刻把这一招用在自己的网页上？

仅仅只在它真正会起作用的时候。

环境提示这一概念在互联网上并没有完全推广开来，而现实中却存在着各种各样的应用方法。你应当开阔视野，看看身边有什么能够激发灵感，提升自己的设计。如果能够雪中送炭地帮助用户找到所需的信息，那你简直太棒了。

不用强迫自己这么做，但是请时刻留意去寻找机会，同时也要留意去发现灵感，哪怕它们看上去和你所做的事毫不相干。天知道它们什么时候就会让你灵光一闪，找到一种更新颖、更有效的设计思路。

在这个案例中，我做的一些设计较之以前的设计确实改进了不少，而我的出发点仅仅只是为用户提供导向。为创建明确的路线而做的任何事情对用户都是有益的。他们只需要扫一眼页面就能自动理解链接顺序的含义，只凭直觉，而不需要思考和检验。

第 5 章
标签云里抬头看路

在 *Designing the Obvious* 书中我曾提到有位设计师带着他设计的博客模板来找我。这个设计里面有一个带有标签云的工具栏，此外还有两个按钮：一个是按标签云显示，另一个是按列表显示。

在这里我想把那个故事稍稍再扩展一下，因为自从写了 *Designing the Obvious* 之后，我又学到了一些东西，它们改变了我之前的观点。

标签云是一种链接列表，显示在一个区域内，并且按照热门程度（或者其他准则）来加重显示其中的某些标签。最后的结果看上去就像一片链接的“云层”，就像 Flickr 网站所设计的那样[1]。见图 5-1。



图 5-1

① 示例见 http://www.flickr.com/photos/tags/。——译者注

Flickr.com[①]是第一个使用标签云的大型网站，而这个想法也早已为无数其他网站所借鉴。所以当这位设计师与我探讨博客模板时，我想的是："酷！咱们就这么干。标签云现在越来越流行，让我们看看这里是否也能用上。"

那位设计师根据自己的想法已经做了非常深入的设计。最开始他建立了一个简单的链接列表，并且按照链接所指向的内容热门程度进行了排序。如图 5-2 所示。



图　5-2

下一步，他通过某种程序算法调整了列表中每一个链接的字体大小。链接所指向的内容越受欢迎，该链接的字体就越大。为了强调字体大小不同，他把列表按照字母顺序重新排序，见图 5-3。



图　5-3

① Flickr.com 是一个以图片服务为主要的网站。它提供图片存放、交友、组群、邮件等功能，其重要特点就是基于社会网络（Social Network）的人际关系的拓展与内容的组织。这个网站的功能之强大，已超出了一般的图片服务。网址为 http://flickr.com/。——译者注

然后，他去掉了列表项之间的断行，使它们排列在一起成为如图 5-4 所示的一片“标签云”。



图 5-4

由于这个系统既能支持按标签云的方式显示，又能按普通列表的方式显示，因此最后他在界面下方加上了一个 Cloud 按钮和一个 List 按钮，使之可以在这两种视图之间相互切换。见图 5-5。



图 5-5

乍看上去这是个很好的主意。

5.1 你的创新是否有效

这个设计的创新之处在于它提供了两种视图。在此之前，别的网站都没有在标签云中进行过这种尝试。而且这位设计师赞同的这个想法是源自这个模板开发者的意见。

“程序员告诉我用户可能会希望切换视图。”他说。

经过一番考虑，我认为应该更注重于提高品质，而不是一味地创新。与其硬要在标签云里强塞入一些所谓的新贡献，还不如让它就那样好好呆

着。我们最起码已经把链接列表改善成了标签云的形式，不是吗？这对于提高品质来说已经足够了，况且添加几个按钮也有些画蛇添足。

我想那些知道标签云是怎么回事的人会从这种视图中获得方便，而那些不理解标签云的人所看到的仍旧是一个链接列表。这种视图并未使任何人受损，也没有违反规则。

我不认为用户会在页面中浪费时间——瞪着工具栏看半天，然后琢磨到底是该用标签云还是列表。这里并不是必须得遵循用户习惯方式的重要特性，它只是一个简单的链接列表而已。所以我告诉那位设计师去掉那些按钮，只显示标签云就行了。

我相信我的观点是正确的。但同时我也相信当时自己考虑得还不够透彻。

5.1.1 什么时候该说不

很有趣的是，标签云和之前“环境的提示”（我们在第 4 章讨论过）的概念有一点点背离。单纯从技术上来说，它们起到的功能和第 4 章中排序的链接列表一样，通过视觉上的不同来传达更多的含义。但是使用标签云需要一个前提，而第 4 章的排序列表却不需要。

那就是知识。

如果真正想从标签云中获得方便，就需要先弄明白为什么有些链接比其他的要大，或者比其他的颜色深。而这些问题就颇值得商榷了。

我观察过的绝大部分用户在碰到标签云的时候，都不明白为什么有些链接比其他的要大。只有极少数的人才会去猜它的原因，而其中又只有更少的一部分人真正在乎这一点。而即使单击了某个标签也无法增进他们的理解。（顺便说一句，以上是我非正式的观察结果，非常欢迎不同的观察结论。如果我错了，请务必告诉我！）

这个事实意味着你完全可以抛弃标签云的形式。毕竟，用户也确实没有必要理解它蕴涵的逻辑意义。他们最终还是得到了一个链接列表，不是吗？

从技术上来说，是这样的。但是标签链接一般来说都很短，基本上只包括一到两个单词，而且很少能把它们自身的含义表述清楚。

以上所有这些事实告诉我们，几乎没有多少人能真正明白标签云的含义。

我和那位设计师说用户不会花时间关心如何个性化显示那一组链接，我只对了一半。

当然，他可以在这个标签部分加上明确的标题，例如“最受欢迎的主题”。他也可以加上一些说明性文字告诉用户链接所指向的内容越受欢迎，该链接的字体就越大。

但是，我应该告诉他直接把整个标签云去掉，仅仅显示一个短列表，按照受欢迎程度排序即可。为什么呢？

因为在这个世界上不存在任何理由强迫用户思考，无论对于一个页面、一个场景，还是一个博客模板。

我们没有理由要求用户学习了解为什么有些链接比其他的要大。用户也没有理由被迫花精力去学习标签云，更不用说去探究它的重要性了。

Flickr 的用户经常会上传大量图片以便和家人及朋友们分享。而在 LibraryThing[①]，一个图书编目网站，用户会为自己的图书归档并编制目录。在这些例子中，标签云确实有助于用户学习曲线的升高。它能让用户粗略地看到人们是如何使用网站的，并且为他们快速地提供评估信息。但是对于博客来说，大部分浏览者只会短暂地逗留，看一两篇日志，然后就离开。所以，博客绝对不是一个强迫用户学习新潮交互方式的好地方。

如同刚才所说的，用户只会在页面短暂地逗留，而不会考虑如何个性化设置链接的显示方式。我也从来没看到过哪位用户会在第一次碰到标签云的时候花时间学习它的功能特性。往好了说，它会迫使用户在不必要的地方思考；而往坏了说，用户很可能因此而困惑，甚至烦恼。

这才是我当时应该告诉那位设计师的结论。

不过我想，我现在已经告诉他了。

---

① LibraryThing 是一个专注于书籍的社会化网络，在功能上和国内的豆瓣早期有些类似。该网站有着比较出色的书籍检索功能，用户能彼此分享读过、正在读或喜欢的书籍。网址为 http://www.librarything.com/。该网站的标签云页面见 http://www.librarything.com/tagcloud.php。——译者注

5.1.2　如果你不是解决方案的一部分……

前几章我们介绍了页面布局、品牌商标、站点导航，甚至还包括如何创建环境的提示来协助用户获得导向。

然而，创造良好的第一印象还需要时刻知道什么该展现，什么又不该展现，要有取舍。当时那个博客模板的标签云就是由于我自己所导致的失误，而它对今天的用户可能仍旧有影响。不过，尽管这件事让我很难受，但我意识到一个人是不可能知道所有事情的。设计是一个永无止境的学习过程——总有更多的东西得去认识——而且不管我们有多么渊博，有时难免仍会误入歧途。

最重要的是我们需要不断地自我检验做事情的方式，尽可能确保所做的设计正逐步向自己的预期靠拢。

设计绝不包含任何运气成分。一个网站的外观、版式布局、导航结构、传达个性的元素，这其中的任何一项，乃至颜色和字体的使用方式，对于第一印象（以及随后的效果）而言都至关重要。它们都需要我们有明确的目标，进行针对性的设计，才能传达正确的信息，而用户才会信任我们的网站和产品。

我们将在第 6~8 章中看到，仅仅这些还远远不够。即使如文本那样简单的东西，也同样会影响用户对我们的看法、对产品的理解，以及他们在网站的逗留时间和对产品持续使用的时间。

再怎样强调都是不过分的。设计中不要留任何漏洞。

每件事都很重要。

第二部分

学　　习

你是否见到过那种占据网页顶部大量篇幅，却又臭又长让人根本不愿理睬的大段文字？是否遇到过要求你确认是否取消操作的对话框，却被迫在“取消”和“确认”两个模棱两可的选项间徘徊不定？是否遇到过那种连一点说明都没有的复选框，而必须自己去揣摩、猜测各个选项的意思？

绝大多数网络用户都碰到过这些情况。

这些文字说明和界面中其他任何元素同等重要。而且在大多数情况下，用户必须依赖这些文字说明。

在试图理解某个网站，或者考虑如何完成某项任务的瞬间，页面中的文字将直接影响我们对这个网站的信心。结果是我们要么爱上这个网站，要么从此敬而远之。

好的文字可以解释开发者的用意，说明产品的好处，还能介绍使用方法；反之，不好的文字则会把用户搞得头昏脑胀，被迫靠猜测做决定，这很可能会使他们犯下错误，然后责怪自己。

人们使用我们的产品时，他们所知道的一切都来自于我们所告诉他们的东西。因此，我们的职责是给出正确的信息，而且要尽可能简明清楚。哪怕是简单到把名词改成动宾词组，或者是开门见山的几句话，都将会深刻地影响用户对我们网站的看法。而当文字无法达到目的时，我们还可以通过其他方式来解决问题。

在下面的 3 个故事中，我们将探讨如何更清晰明了地与用户沟通。首先通过头两个故事讨论如何修改文字以便传达更多的含义，而在第三个故事中我们则讨论当文字无法胜任时，如何使用视频来弥补。

第 6 章 开门见山

我曾经与一个小型的在线教育软件公司合作为某家知名航空公司开发一套学习管理系统（Learning Management System，LMS）。这套系统能让该公司各机场数以千计的工人随时学习并了解最新的安全措施，同时掌握新技术以提高自身的技能水平。

这些工人中绝大部分都没有接受过大学教育，因此人力资源部门认为所有的内容都必须为他们提供透彻的解释。尽管这个想法颇有些自以为是的施舍意味，但是在评审设计的时候它反而帮了我的忙。

6.1 为什么说欢迎辞没有用

该系统首页设计包括一大段文字作为欢迎辞。它解释说明用户可以在线指定课程，从而帮助他们学习新知识，并且了解最新安全规章。见图6-1。

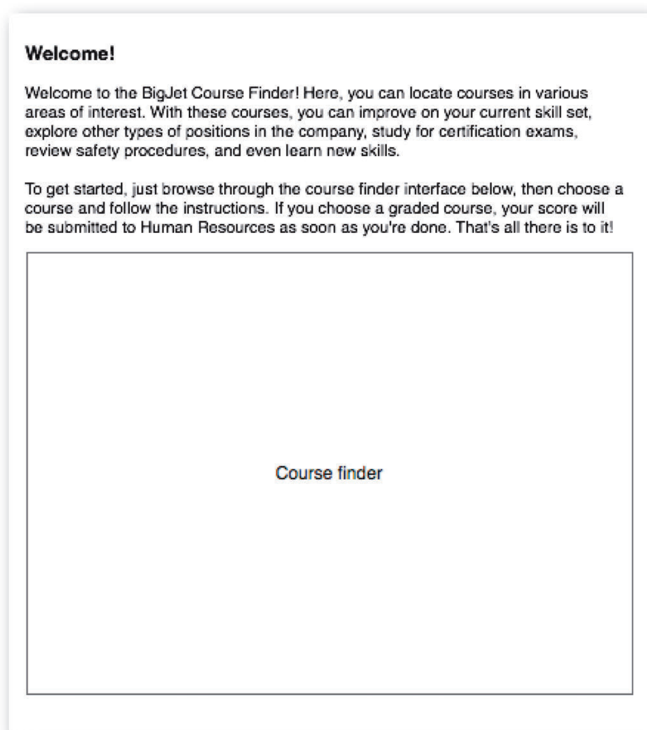


图 6-1

我敢肯定你在网上看到过类似的信息起码有一百万次了。但有多少次你去读过呢？

Jakob Nielson 在他的 Alertbox 专栏[①]“废话文本：保留、删减还是去掉？”[②]一文中指出，人们通常不会在网页上阅读文字，他们只是扫描。如果是更长的介绍性文字，人们往往会直接忽略。也就是说，篇幅短的文字才有可能吸引一部分注意力（假设它真的有用），从而有助于用户理

① Jakob Nielson 是尼尔森 - 诺曼集团的主要负责人之一，在 Web 可用性领域内被公认为世界的权威。他的作品很多，包括畅销全球的 *Designing Web Usability* 一书。他的 Alertbox 专栏于 1995 年开始在互联网上发布，网址为 http://www.useit.com/。——译者注

② 这篇文章英文名为“Blah-Blah Text: Keep, Cut, or Kill?”，原文见 http://www.useit.com/ alertbox/ intro-text.html。——译者注

解他们正在看什么。

在大信息量的站点里我们会专注于阅读，然而当探索一个新的 Web 应用程序时则与之相反。我们会寻找那些显著的线索，目的是为了了解这个网站是干什么的，它是如何运转的，以及弄明白要想使用它，我们需要做什么事情。换句话说，我们考虑的是做，而不是读。我们希望将事情搞定，因此采取的每一个步骤都是为了完成某件事情，不管这件事情是什么。

比如说，看到这个欢迎辞的时候，我立刻绕过它，把注意力集中在课程列表中去。那才是这个页面最主要的交互元素。

6.2　为扫描而设计

不过最后我还是扫了一眼欢迎辞，因为评审这个设计是我的工作。但是当我扫过这些文字的时候，看到的是如图 6-2 所示的内容。

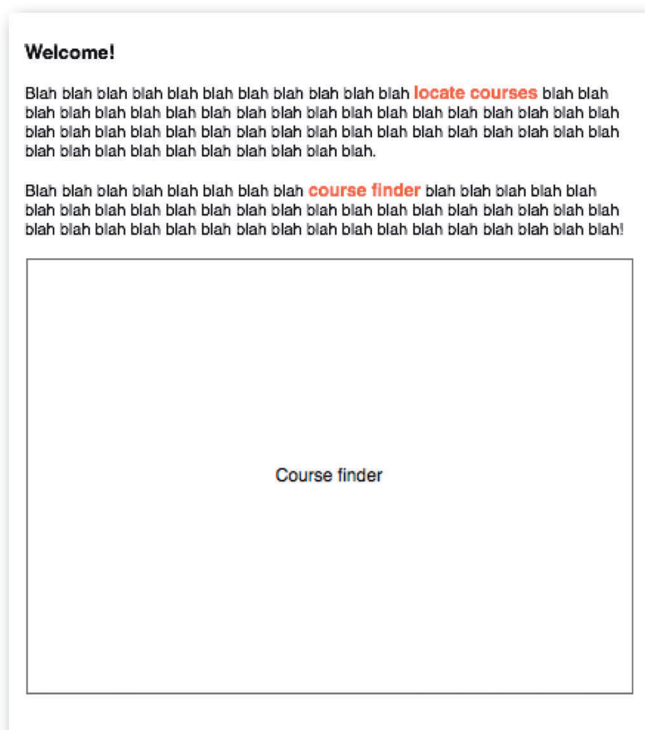


图　6-2

你阅读欢迎辞时所看到的内容大概与此非常相似。我们更感兴趣的是要完成什么事情，而不是阅读一段毫无目的、对达成目标毫无帮助的文字。

第一个问题就出在这段文字的标题是“欢迎”，这就好像公司在某个节假日派对上打开办公室的大门迎接客人。但是这里不是个节假日派对，而是个网站。浏览者们也并不是客人，而是有好多工作要忙的人。他们为了完成公司规定的安全课程中断自己的正常生活，或者为了获得升职的机会而被迫学习某项新技术。

他们并不关心自己是否受欢迎，他们关心的是找到某个课程。

所以首先，让我们给他们一些自信，把标题改成 Find a course（找到你的课程）可以让他们知道自己打开了正确的页面。见图 6-3。

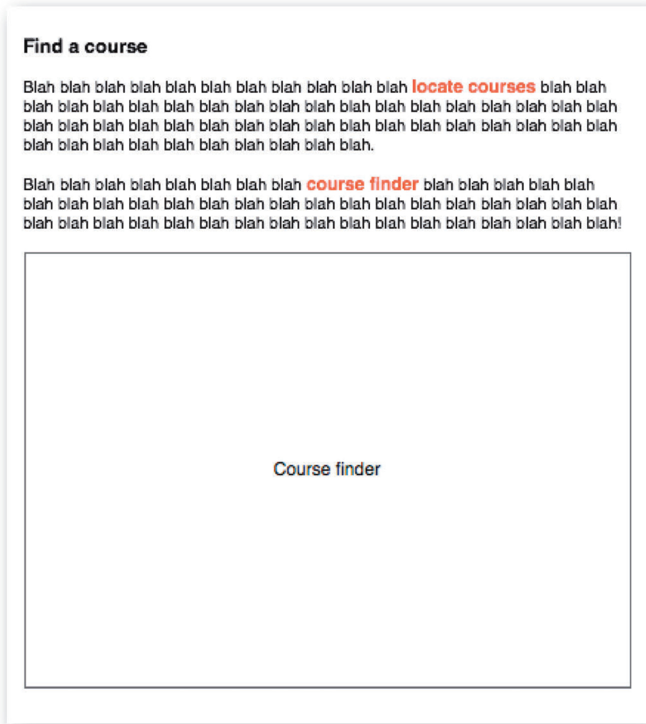


图　6-3

好多了。但是那一大段愚蠢的文字还在那儿，我们来把它变得更有用一些。

“赶快行动”词组

“赶快行动”词组是用来激发人们做出某种行为的一组字词。它近似于一种命令，比如“回你的房间去”，但又不会那么缺乏友善。

你曾经见过它们，可能也不少于一百万次了。它们一般被陈述成这样：“保存更改”、“阅读更多”以及“登录网站”。

它们还易于扫描。因为使用动宾词组能够准确地告诉用户即将发生什么，同时还能让他们感到自己才是软件的指挥者。

使用 Find a course 作为页面标题能够帮助用户了解在这个页面上他们要干些什么，但是那一大堆文字却没能提供多少具体的内容。

因此下一步我们需要去掉这些文字，然后用更简练、明确的话替代，并且在里面包含“赶快行动”的陈述。

在这个案例中，要想解释页面的主要目的，我要使用的文字连一个段落都不到。很多时候所需要的仅仅只是一句话而已，在解释你在页面中能干什么的同时就能说明它的目的、好处和使用方法。

经过慎重考虑——花了大概五秒钟的时间——我开始设法扭转现有的局面：把两大段文字的内容全部去掉，然后写上很简单的一句话。见图 6-4。

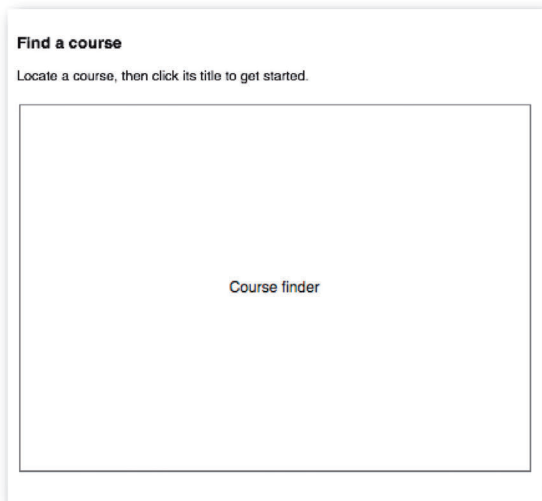


图 6-4

非常好。做到这样也并不困难，不是吗？

新的页面版本直入主题，这样我们不会浪费用户的任何时间。

但不幸的是，事情并不那么简单。

客户不愿意放弃欢迎辞里面的文字。很显然，市场部的某个人为了它殚精竭虑，而且确实花费了相当多的时间考虑如何让措辞完美。

“如果用户需要更多的信息怎么办？”他们问道。

“他们不会。”我回答。

这种争论似乎无法得出合适的结论。

最后结束这场争论的却是这样一句话：

“这些人只有高中文化水平，如果他们读了这些我才会吃惊呢。”（嘿嘿冷笑）

该公司人力资源部门对自己员工智能水平的判断，我对其完全持保留态度。然而这样一来事情变得好像缩减文本成了他们的主意。

他们采用了我的设计。而且据我所知，该网页从没有出现过人们不知如何使用或者不明白它能做什么等方面的问题。

从这一课我们能学到什么？

对用户来说，一个简短的句子远比一大段文字要来得有意义，即使这段文字的文笔很好。重要的并不是我们提供的信息量有多大，而是我们能否给他们提供真正需要的信息。

第 7 章
为界面做标记

最近我为一家在线教学软件公司的产品做可用性评审。该产品是一个基于 Web 的课程创建工具，已经是第二个版本了。

这款产品主要为在线教学系统的开发者设计。比较熟悉在线教学课件以及成绩跟踪系统的人可以通过这个工具选择模板、创建幻灯片，从而在较短的时间内创建好自己的课程。用户要做的是开设某个新课程，创建所需要的幻灯片并为其选择模板，然后填写一些表单以便为每张幻灯片添加内容。

当我评审这个应用程序的时候，脑子里经常出现一个念头，认为它完全可以面向更多的受众群体。只需要一些相对简单的调整，这个产品就能适用于培训设计师、人力资源部门以及更多的人，而不仅仅只是面向熟知在线教学课程开发的专业人士。

自然，扩大了用户群也就意味着能为公司带来更多的利润。不出所料，客户非常赞同这个想法，因此我把自己的评审意见更多地集中在如何实现这一点上面。

由创建课程的界面开始，在这里用户需要填写课程 ID 和标题，然后选择一系列选项来建立新课程。如图 7-1 所示。

图　7-1

因为我完全不知道应该如何填写有效的课程 ID，所以我输入了 My Project。完成表单之后，得到了如图 7-2 所示的一个错误信息。



图　7-2

噢，谢谢。但是为什么它会无效呢？我怎样才能使之有效？有效的课程 ID 看起来应该是个什么样子？

这个界面“假设”我具备很多相关经验。它里面所有控件的标记都是基于这种“假设”之上的。

7.1　停止标记你的“假设”

互联网上存在着无数缺少上下文语境的输入栏、复选框和单选按钮。它们要求用户输入、选择，却不解释这些行为的含义。

我们经常自以为是地猜测用户对产品的理解程度。当我们在某个复选

框后面写上 Persistent index（持久索引），或者在某个输入栏旁边写上 Newproject ID（新课程 ID）的时候，总是在假设用户会明白我们的意思。我们先做假设，然后去标记这些“假设”。

就在这个界面中，其实我们是在要求用户为了使用我们的软件，他们必须知道一些只有我们才知道的事情。我们要求用户必须先揣测我们的心思。

如果用户失败了，我们还经常因此而责怪他们。难道这不是很搞笑吗？

现在告诉你如何修正这个问题。

检查应用程序中每一个控件的标记，并且问问自己，一个从没见过这个程序的人是否也能明白它的意思。把所有回答是“不”的项目列一个清单，要么修改这些标记使其变得通俗易懂，要么加上说明元素（文本、屏幕截图、小短片等）解释它们。

把所有回答为“是”的项目同样也列一个清单，并把这个清单给 3 个从没接触过这个程序的人看。

根据他们看的情况，估计你还得再改几条。

不过，用不着难受。今天早些时候我自己的列表里面都至少改过 3 条呢。

至于前面说的可用性评审，一开始我为课程 ID 提出了简单的改进建议，只是加上了一点文字，显示有效的课程 ID 应该是什么样子。见图 7-3 左侧。

图　7-3

第一步迈出来之后就很简单了，我只需要把这个想法扩展到界面其余的控件上面去。

首先，为大部分控件加上相关帮助文档的链接。其次，为各输入栏加上说明性文字，解释什么样的输入才是有效的。最后，修改几处标记，使它们的意义表达得更为清楚。例如，Transition time（切换时间）变成了 Transition time，in seconds（切换时间，单位：秒）。见图 7-4。

图　7-4

设计不能只基于我们自己一厢情愿的“假设”。相反，我们需要假设用户毫不了解我们的软件，随后给他们足够的信息以便帮助他们理解。

说明性文字、What’s　this?（这是什么？）链接以及表意清楚的标记都能达到这一目的。

第 8 章
生动传神的视频

我大部分的工作都是在指导别人如何能让事情变得一目了然。但是偶尔我自己也会有那种突然被闪电击中的感觉——有些过于明显的东西我却一直视而不见。

有一次我为某款产品做可用性评审，就发生过这样的情景。

那天我发现自己正在绝望地试图用书面文字来解释在某个表单中一个输入栏的问题：它没有向用户说明自己有字数限制。这个输入栏里面只能输入 50 个字符，但却没有相关的说明。因此当达到字数限制时，新输入的字符就不再显示了。这看上去就好像我的键盘突然失灵了一样，我敲击键盘没有任何反应。

那天我绞尽脑汁想把措词整理得顺当一点，以便把这个现象说清楚。我想到过是否该使用一两张屏幕截图，但那又有什么用呢？客户通过这些截图所能看到的只是一个里面满是字符的输入栏而已。它们无法表现出那种键盘失灵时的无力感和挫败感。

我又想是否可以把每一步都写出来，把整个过程描述一遍。但是那样似乎工作量会很大，而且如果连简单的措词都不能使自己满意的话，就更不能指望写一大堆步骤能好到哪里去了。

而就在这时那道闪电出现了。

视频，你这个白痴。视频！

8.1 视频演示胜过千言万语

视频演示片段，也就是大家所说的 Screencast[①]，现在已经应用到各种网站中去了。这可能是因为创建网络视频文件变得越来越容易吧。

比如在 Basecamp[②]（由 37signals[③]开发）里面，每一个“空白页”（用户尚未创建内容的页面）都提供了一个 Screencast，如图 8-1 所示。

这就是 Screencast 其中一项伟大的用途，而且相当到位。在用户需要的瞬间，他们能很快地学会如何使用这个程序。我经常使用 Basecamp 来管理客户项目。大多数客户都是第一次使用 Basecamp，但是有这些视频能帮助他们很快上手，这真是一件令人安心的事情。

而超越这一功能并且名声大振的则是 Common Craft Show[④]里面创建的视频。这些短片本质上来说并不是 Screencast，因为它们并不是电脑屏幕录像，但却能非常生动、活泼地介绍一些新的或经常为大众所误解的技术。

图 8-2 中的这个视频探讨的是小型节能荧光灯管。Common Craft 也创建过关于维基百科、RSS，甚至在线图片分享等方面的视频。

---

① Screencast 这一网络术语尚无一个公认的中文译名。所谓 Screencast 指的是一种个人使用屏幕录制软件制作的视频文件，通过某种方式（通常是 Flash 格式文件）放到网上共享，以达到娱乐或者教学、示范的目的。其在英文维基中的解释如下：“Screencast 是一种计算机屏幕输出的数字记录，也被称为是屏幕录像视频。它通常也包含一些叙述性的音频。尽管 Screencast 这个名词在 2004 年才出现，但如 Lotus ScreenCam 般的产品在 1994 年即已开始使用。早期产品生成很大的文件，对其进行编辑也很受限制。但如今已能支持更为简洁的文件格式，如 Adobe Flash，并且支持更多的编辑操作，例如改变顺序、鼠标路径追踪，以及音频等。如果说屏幕截图是用户屏幕的一张静态图片，那么 Screencast 本质上就是用户在他们监视器上所看到的一段影片。”具体页面见 http://en.wikipedia.org/wiki/Screencast。——译者注

② Basecamp 是一个基于 Web 的项目管理和协作工具。用户可以通过互联网写作并分配任务、发表意见或反馈、安排日程、网络共享文件等。网址为 http://www.basecamphq.com/。——译者注

③ 作为 Web 2.0 的经典网站，37signals 因为开发了 Ruby on Rails 和一系列基于 Web 的工具而闻名天下。网址为 http://www.37signals.com/。——译者注

④ Common Craft Show 是 Common Craft 公司的网络展览项目。该公司位于美国西雅图，只有夫妇二人。如他们自己所说，公司的产品就是“解释”。该夫妇使用白板、油性笔和纸片制作各种视频短片让复杂的事物变得容易理解。网址为 http://www.commoncraft.com/。——译者注



图　8-1



图　8-2

这些视频其实需要艰辛繁琐的制作过程。制作者需要在白板上创建一系列图画，并且为这里面的每一幅图画的每一个步骤进行叙述性的拍摄，此外讲述者还需要手绘一些漫画作为补充。这些视频的目的是讲述那些技术是如何工作的并且给人们带来了什么好处。

8.1.1　使用视频来表述问题

突然想起用视频来传达信息的种种好处，我意识到自己也可以用一个短片来向客户解释输入栏的问题，这样就不需要再花精力去说明了。一目了然，不是吗？

因此我马上重新开始，再次尝试在输入栏中敲入 50 个以上的字符，并且把整个过程的屏幕都录了下来。与此同时我也录制了声音，这样客户就能在看的同时听到我无济于事地敲击键盘。见图 8-3。

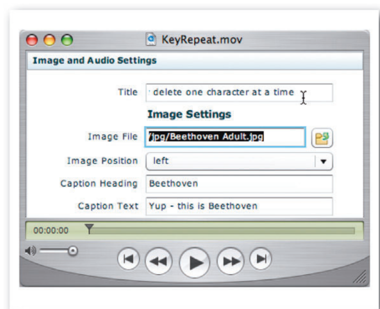


图　8-3

我甚至还加入了一点口头叙述以便表现出在这种情境下用户可能会想什么。

“我的键盘……坏了。没反应了。我不知道为什么。也许它的接口松了……不是。也许我是个智障……不是，我是个 MBA。也许我需要重启电脑。太没劲了——我才刚刚喜欢上这个网站。噢，好吧。”

20 秒钟后，一切搞定。我重播了一次，以确保录制效果不错。就这样，灵光一闪不到一分钟就解决了问题，而且比一开始预料的要生动有效得多。

当我把这段视频和其他评审意见发给客户之后，他们明白了我的意思。你很难和视频去辩论什么。

8.1.2　使用视频来表述想法

就在这次小小的成功之后，我意识到如果能用视频来描述问题，那么也同样可以用它来描述那些想法，这和用原型（Prototype）来描述是一样的。视频能直接放到客户面前让他们看，然后让他们思考，然后再看一遍。而不需要任何解释。

我可以录制一段视频来说明某个交互应当怎样发生才最为合理，并且加上少许叙述，而不用把这些过程全都写出来。这就像一个原型的视频版本，或者说一个 Screencast 的原型版本。随即一个词跳入我脑子里。

天才。天才啊，伙计！如果你此刻听到一声小小的“哇哈哈哈哈”，那其实是我为自己的才华得意大笑的回声。

一开始我就像个白痴，但现在我得到了补偿，而且是双倍的。

不管怎样，我立刻找到了这样做的全部理由，并且在我后来呈交给客户的设计文件中都会包含一些视频演示片段。这些简单的屏幕录像节省了我大量的时间，而且在绝大部分情况下，它们所传达的内容比之前那种一步一步的文字案例描述要生动、丰富得多。

8.2　极为简单的视频演示艺术

为网站创建视频演示片段或者说是 Screencast，是一个很简捷方便的过程。只要你有合适的工具，并且愿意花几分钟时间做准备。

首先，搞清楚你要传达什么内容。在大多数案例中，我只想展示交互动作应当怎样发生，因此我所需要做的就是展现一系列界面的屏幕状态，并且解释在每一个界面中用户应该做些什么才能到达下一个屏幕状态，最后完成交互。

举个例子，为一个五星级的界面做视频演示的时候，你需要展现的部分包括：默认状态、单击哪里来设置评级，单击之后屏幕上会发生什么、如何完成评级，以及完成后屏幕是什么样。这个过程看起来很明确，但这同时也意味着需要通过某种方式来创建一个简单原型，这样才能通过单击进入各个状态。

我更喜欢 OmniGraffle[①]，因为我经常使用它并且非常方便。当然你也可以使用 Adobe Flash、Microsoft Visio（只适用 Windows 系统）、Gliffy[②]（基于 Flash 的 Web 应用程序，与 Visio 很相似），甚至 Microsoft PowerPoint。

例如在 Flash 中，你可以创建一系列帧，每一帧都显示一个不同的屏幕状态（使用线框图或者高质量的设计图稿），然后为每个可点击的项目简单地创建按钮动作。最后发布文件并用屏幕录制工具记录下完成交互的每一个步骤。

---

① OmniGraffle 是 the Omni Group 公司的产品，现已发布至第 5 个版本。它是一款流程图及界面原型绘制软件，界面美观，能支持 Visio 文件格式。但只适用于 Mac 系统。网址为 http://www.omnigroup.com/applications/omnigraffle/。——译者注

② Gliffy 是一款在线绘图工具，用户能使用该软件绘制 UML 建模序列图、流程图、组织图、室内设计图和网络拓扑图等，功能强大。绘制好的作品可以保存并发布到自己的 Blog 中。网址为 http://www.gliffy.com/。——译者注

录制屏幕时可以使用各种各样的工具。我使用的是 Snapz Pro X[1]（只适用 Mac 系统），录制影片时就像屏幕截图一样简单。Windows 用户可以选择 TechSmith 公司的 Camtasia Studio[2]，操作方式大同小异。

最新也最酷的屏幕截图和录像工具则是 TechSmith 公司发布的 Jing[3]。这是一款免费软件，见图 8-4。

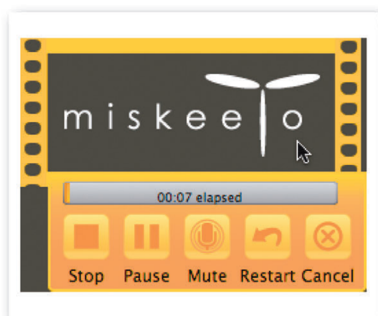


图　8-4

单击几次就可以创建屏幕截图或者视频，而得到的文件可以通过软件自动赋予的 URL 地址或者 Screencast.com[4]账号与所有人分享。

换句话说，录制完一段影片之后，你可以单击共享图标把该影片上传到 Screencast.com 中去。你可以粘贴它的 URL 地址（当视频上传后这个 URL 会自动复制到你的剪贴板中），也可以保存影片过段时间再发布。

运用文字显得困难时，考虑使用视频来帮助你的用户快速上手吧。

噢！还有，如果碰巧你在我之前已经想到了视频演示这个主意……别让我知道。我宁愿继续相信它是我的独家发明。

谢谢。

---

① Snapz Pro 是一款全能的屏幕截图软件。用户可以设定全屏、当前窗口、自选区域范围截屏。其中全窗口截屏功能可以截取任何窗口的完整内容，而无需使用滑动条重复截取不同部分再用 Photoshop 拼接。这款软件能将屏幕动作录制为 QuickTime 影片，并能同时记录系统内的数码音频信号及麦克风输入的音频。网址为 http://www.ambrosiasw.com/utilities/snapzprox/。——译者注

② Camtasia 能在任何颜色模式下轻松地记录屏幕动作，包括影像、音效、鼠标移动轨迹、解说声音等等，还能对视频片段进行剪接，添加转场效果，并可将电影文件打包成 EXE 文件，在没有播放器的电脑上也能播放。网址为 http://www.techsmith.com/camtasia.asp。——译者注

③ Jing 能够支持 Mac 及 Windows 系统，在 Windows 下则需要 .Net 3.0 组件支持。Jing 集成了 TechSmith 的另外一项服务 Screencast(专门存储截屏文件的空间），用户可以在截屏或录制后直接把文件上传到 Screencast 中，同时还能像 YouTube 那样通过代码调用把视频内置到自已的 Blog 里。网址为 http://www.jingproject.com/。——译者注

④ 该网站网址为 http://www.screencast.com/。——译者注

第三部分

搜　　索

把被动的网站浏览者转变为主动的活跃用户其实很简单，只需要把他们的目标从浏览转变为搜索。而且搜索是人们在互联网上最基本而又最重要的任务之一。Google 告诉我们，好的搜索引擎可以让人们在数秒钟之内找到任何方面的资料——没有做不到，只怕想不到。

当然，并不是所有的搜索引擎都像 Google 那么好用。事实上，有很多搜索引擎在提供信息入口方面都做得极差。这并不一定是因为引擎的算法问题，其实有些算法相当不错。

问题经常出在界面上。

事实是这样的：绝大多数使用过搜索引擎的人都用过 Google，而绝大多数用过 Google 的人都把它当作是完美搜索的黄金标准。如果不是 Google，那么就是垃圾。

在这种情况下，我们该如何竞争？我们该怎样才能为用户提供他们想要的信息，而不是眼睁睁看着他们离去然后打开 Google ？我们该怎样帮助用户快速找到正确的信息？而在用户看到搜索页面或搜索结果列表的时刻，我们又该怎样帮助他们快速上手？

为了回答这些问题，我要讲 3 个故事，分别从不同的侧面探讨搜索。第一个故事是关于搜索本身，第二个与搜索结果列表有关，而第三个则是如何精炼搜索条件以便得到更有效的结果。

当然，所有这些故事都是在讲述如何创造各种积极的瞬间。它们会让用户留在你这里，而不是跑去其他什么网站。

第 9 章
提出搜索词建议

2006 年，DoTheRightThing.com[①]（缩写是 DTRT）网站发布前不久，DTRT 的创始人 Rod Ebrahimi 和 Ryan Mickle 找到我，想让我为他们的设计做一些可用性评审。而我在那一刻一定感觉自己非常高尚慷慨，因为一听到他们的主意，我就宣布无需他们支付任何酬劳。简单来说，我真的很欣赏这个想法，而且我愿意为之贡献自己的力量。

Rod 和 Ryan 试图要为世界带来一些改变。他们建立了一个主页，里面都是由用户自发创建的内容，全都是关于各种公司或团体的行为及举措。比如说，像 Adobe 或者 Yahoo 这样的公司如果有了什么善举，网站的成员就会发布文章介绍这种行为，以图引起公众注意并且让其他公司仿效，去“做正确的事”。同样地，当某些公司做的事不够体面，这些文章也能发动大众进行声讨。网站成员也能为每一个故事评分，以此表现出该事件所造成的社会影响力。

这真是个高尚的想法，而且我很骄傲自己能为 DTRT 出一臂之力。

然而在他们能够撬动地球之前，他们迫切地希望能找到一个支点，即要改变网站中最重要的一项任务流程：怎样发表一个新故事。

在做评审的 3 天时间里，我一直在想应该如何发表新故事，但总是得不到一个令人满意的答案。

最初的设计版本是，在发表关于某家公司好的或者坏的故事之前，你

① DoTheRightThing.com 这个网站的中文名称可以是“为所应为”，或者说“做正确的事”。在这里社会大众可以对一些公司进行自由评论。人们可以共享对各公司的发现和认识，直接了解各公司动态，也能实时获取并评论各公司在社会上所展现出的形象。网站里也有关于中国的文章，例如“星巴克是否应该搬出紫禁城”。该网站的网址为 http://www.dotherightthing.com/。——译者注

必须单击 Companies（公司）标签，然后选择这家公司的名称。

这个设计且不说效率低下，它让人很难弄明白到底该怎样才能发表一个新故事。最终，你必须进入该公司的详情页面才能做这件事情。而当你想要发表某个故事时，没有任何理由需要跑到那个公司的网页上去。为了让这一过程一目了然，需要通过某种方式让我们在网站的任何地方都能发表文章。

我建议在网站的每个页面上都提供 Submit New Story（提交新故事）按钮。然后在之后的页面中，用户可以从列表中选择某家公司，也可以自行输入新的公司名，然后直接发表自己的故事。

DTRT 很喜欢这个主意。让我感到特别高兴的是，两个星期之后他们给我展示了新的解决方案，给了我一个大大的惊喜，他们的设计比我预期的要好得多。

他们并没有简单地往页面上丢一个公司列表，然后让用户去单击选择，而是采用了一种搜索功能。更准确地说，是一种能够“自动完成”的搜索功能。

9.1 运用 Poka-yoke 机制，使用自动完成输入框

Poka-yoke[①]是一个日语词汇，有“**防止错误**”的意思。人们通过这种 Poka-yoke 机制侦测错误，或者防止错误发生。

而“**自动完成**”的设计思路就是 Poka-yoke 预防机制的一个例子。

“自动完成”指的是某种奇特的文本输入框。它们能够尝试猜测用户输入的文字，并且动态地搜索出适配的结果，然后推荐给用户，辅助他们完成输入。有些输入框会直接在框内显示最接近的那个词，还有些输入框则会显示一个适配的列表，用户可以在其中选择。

这两种“自动完成”输入框都能避免用户打错字，从而确保他们的搜索

① Poka-yoke（ポカヨケ）一词最早出自于日本丰田（Toyota）公司。Poka 的意思是“错误”，Yokeru 的意思是“避免”。举个 Poka-yoke 的实例，如果自动档的汽车没有挂到停车的 P 档，车钥匙就无法拔出。这样可以防止司机在未锁死车轮的不安全情况下离开汽车。再比如 3.5 英寸软盘的右上角形状有些特别，此举可以避免人们以错误的方向插入软盘。——译者注

更有效率。

而在 DTRT 这个案例中，这种输入框还能在用户发表故事时防止他们为已经存在于数据库中的公司输入各种各样的新名字，如图 9-1 所示。

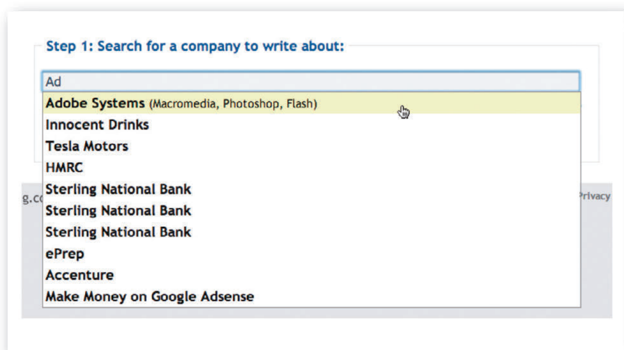


图 9-1

举个例子，如果想写一篇关于 Adobe Systems[①] 的故事，那么用户输入 Ad 两个字母后就会从数据库中搜索并产生一个列表，都是以这些字母开头的公司名。Adobe 碰巧是第一个，因此用户可以直接从列表中选择 Adobe Systems 然后进行下一步。如果 Adobe Systems 没有出现，那么用户可能会输入他认为正确的任何一个公司名称（比如 Adobe），而结果就会导致为同一家公司创建多条数据库记录。

这样一来很可能会出现有好几个页面同时指向一个公司，其中任何一个页面都无法包含与该公司相关的所有故事。

"自动完成"输入框也能预防在搜索中出现一些错误。DTRT 用户经常在只输入几个字母的情况下就能在推荐列表中发现正确的公司名称，而不会因为拼写错误而找不到适配搜索结果。

另外，这种输入框列出的搜索结果也不会分页。因为搜索结果是随着输入的文字增加而变化的——就在当前页面中。因此完全没有必要点开"下一页"，然后再等待其余的搜索结果。

但最重要的是大部分用户很欣赏这种"自动完成"输入框。他们说"它

① Adobe Systems 即我们常说的 Adobe 公司，其英文全称为"Adobe Systems Incorporated"。——译者注

蛮灵的”，而且说这个功能让他们更加信任这个网站，因为开发者做出了相当的努力让事情变得确实简单了。

“自动完成”这种设计方案能带来很多非凡的好处，但它也同样也有一些局限性。

9.1.1　局限性

在 *Designing the Obvious* 一书中，我曾指出 Google Suggest①提供了一种解决方案，它模糊了侦测错误和防止错误之间的界限。

我们发现“自动完成”这一功能其实并不是在任何情况下都能提供帮助。当用户搜索一些小范围的数据时，例如某个自行车零部件的目录，它会很有用，但是对于在诸如 Google Suggest 和 Yahoo Instant Search②这样的网站中进行广域搜索的用户来说，并不一定如此。

换言之，与其说 Google Suggest 教给了我们如何防止错误，倒不如说它在暴露“自动完成”的局限性方面做得更好。

首先，那些盯着键盘输入文字的用户通常根本就不会注意到推荐列表。虽然已经把一切都送到眼前了，他们却还看不到。有些人偶尔会在打完字后抬起头才发现推荐列表，但是他们往往都是敲了回车键后才抬头，所以根本就没看清楚列表里面是些什么。

其次，用户往往会因为推荐而放弃自己先前的主张，但其实他们自己想输入的文字很可能比推荐的词汇更加有效。

比如说，某个希望搜索有关企业内网平台入口的用户可能会希望输入 designing intranet portals 这样的搜索词汇。但是如果在开始输入时他注

---

① Google Suggest 是 Google 搜索引擎最重要的人性化服务之一，即在搜索栏输入框内使用了前文提到的“自我完成”功能，使用户在输入搜索关键词时能及时调整关键词推荐列表，以获得最需要的搜索结果。Google 称其使用了一系列广泛的信息来预测用户的输入内容，例如通过各种搜索请求的流行度来确定这些预测推荐请求的排列顺序等。该功能曾被国外用户评为 2006 年 Google 最出色的新特性之一。网址为 http://www.google.com/webhp?complete=1&hl=en。——译者注

② Yahoo Instant Search 和 Google Suggest 非常类似，在用户开始输入关键词的同时开始搜索相适配的信息。不过它显示的并不是关键词推荐列表，而是在辨识出用户输入的关键词用意后出现一个悬浮窗口，显示相关问题的答案。网址为 http://search.yahoo.com/instant。——译者注

意到了那些推荐的搜索词汇，于是就可能会转而选择其中的 designing intranets，如图 9-2 所示。



图　9-2

之所以会出现这种情况，可能是由于用户认为系统推荐的词汇将会产生更有效的结果，也有可能他们认为这些搜索词汇是搜索引擎开发者希望他们使用的词汇。

第二个理由其实毫无道理。可能你我都知道，但还是有很多人一直都是这样思考的。他们认为当自己被要求或者至少是被引导去做什么的时候，提出该要求的网站肯定比他们自己聪明，因此他们就应该服从。

无论如何，在进行没有固定范围的广域搜索时，“自动完成”功能可能会诱导用户放弃自己本来正确的想法，而去选用网站推荐的，但其实效率较低的搜索词汇。

第三，你必须时刻注意不要让推荐“抢跑”得太快。

在 Yahoo Instant Search 中输入 des 会出现 Wisdom Tree[①]当前财务数据。而在 Google Suggest 里面做同样的事情，最优先的选项则是 Desperate Housewives[②]。但很可能我们想输入的单词其实是 Design。很明显，仅仅只有单词的头 3 个字母，不管是 Yahoo 还是 Google 都不足以马上确定用户所寻找的信息到底是什么。见图 9-3。

① Wisdom Tree（DES）是一家投资顾问公司，它主持开办了一个以货币兑换交易为基础的小型红利基金。与上文有联系的具体网址为 http://www.wisdomtree.com/etfs/fund-details.asp?etfid=15。——译者注

② Desperate Housewives 是指美剧“绝望主妇”。——译者注

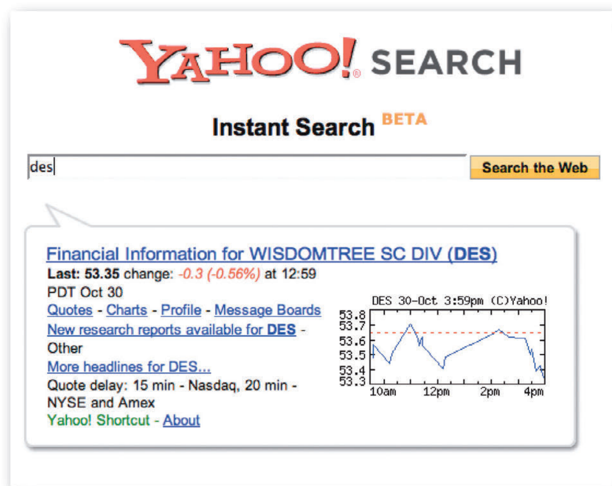


图 9-3

在输入的字符数足以让推荐词有意义之前，请不要显示推荐词。当然从另一方面来说，目前也没有一个魔法数字能确切地规定到底应该在什么时候显示推荐词。因此我们暂且把这个界限定为 3 ～ 4 个字符。

最后，同 Ajax 和 DHTML 开发的任何东西一样，“自动完成”在易用性方面也存在问题。

众所周知，屏幕阅读器是按照从上到下的顺序读取已载入页面的。因此，在页面载入完毕之后“变”出来的任何东西，都需要通知屏幕阅读器页面发生了什么改变，并且在页面的这一部分给它一个快速入口，否则这些部分将无法被视力有障碍的用户“阅读”。

9.1.2 好雨知时节

抛开这些局限性，我们能通过“自动完成”来帮助大部分用户更加快速地获得信息，避免他们在搜索过程中出现失误，防止为同一个数据创建多个数据库条目。同时它还能让你在用户搜寻出信息的瞬间得到他们的赞赏和感激。

对于 DoTheRightThing 网站来说，“自动完成”是一个完美的解决方案。它改变了之前麻烦笨拙的故事发表步骤，加快了该流程的速度并且更为有效。从这一点上来说，他们的确是在“做正确的事”。

第 10 章 列出搜索结果

在因特网上使用搜索引擎时，只需要输入关键词，单击 Search（搜索）按钮之后，我们通常得在海量的搜索结果中努力找寻我们需要的信息。这个行为已经变得很常见，尤其是如今人们每天绝大部分的网络体验都是从 Google.com 开始的。

我们去 Google 搜索并找到某个相关链接，然后单击它。当需要其他信息时，我们再回到 Google。

这样一来，Google 几乎变成了很多人一种根深蒂固的习惯，哪怕是互联网新手也是如此。他们极其虔诚把自己的浏览器主页设置为那个传奇般最精简的网页，并将其作为万事的开始。他们从那里出发去学习新事物、购物，甚至是前往某个熟知的域名网站（是的，很多人甚至可能包括你都是先在 Google 搜索栏内输入某个域名的 URL，然后再单击第一个搜索结果进入那个网站）。

上述的事实有次给我带来了一些麻烦，当时我正在协助某个论坛开发。我意识到如果设计得很差劲，哪怕是简单如一个搜索结果的分页系统，也会让用户痛苦不已。在为该论坛协作开发几周后，我拿到了搜索结果页的设计稿。

而这个分页系统的界面，唔，不是很好，如图 10-1 所示。



图 10-1

首先，当前页面 Page 1 的链接是可以点击的，但它没必要如此。毕竟用

户已经在这个页面上了，对吧？

其次，在这个设计中没有指示能表明用户当前在哪个页面。没有让当前页面的数字凸显出来，事实上它看上去和所有其他的页面链接都是一样的。

与此同时，Page1 中的左箭头也毫无意义，因为在搜索结果的第一页我们无法回退到任何地方。现在的情况是存在两个箭头：向右的可以点击，向左的却不可以。

该部分的标签 Page 也很没必要。它放在那里虽然并没什么坏处，但却显得有点愚蠢，因为它并没有显示当前页数。

那些数字的链接太小了。它们显示的是正文字体大小，但由于每个链接只包含一个字符，因此用户很难把鼠标准确地移动到上面并单击。

最后，这个分页系统放在了搜索结果列表的右上角。这一点打破了包括 Google、MSN、Yahoo 等几乎所有的主流搜索站点设定的惯例。

我现在常常鼓励人们挑战标准，然而至关重要的一点是，在准备挑战标准时请确保你的目标是为了改进它们。如果你的新设计并没有做到改进，那么你的用户将不仅得不到任何好处，反而有可能失去更多。因为他们无法利用在其他网站获得的经验，所以不能很快地理解如何使用你的网站。

10.1　信赖那些行之有效的标准

很多时候，有些事物优秀到足以被普遍接受，好像它本身就是某种标准。只要背离了它，那就一定是不好的。也许该感谢 Google，分页系统也属于这类情况。见图 10-2。

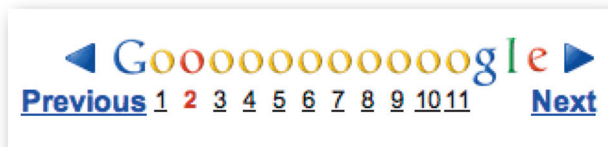


图　10-2

Google 的分页系统非常容易辨识，不仅仅是因为它被 Google 使用，还因为它被无数其他网站所借鉴，其中包括各种主流搜索站点。

不管是在哪个网站，分页系统彼此都很相似。每个分页系统都有一个

Previous（上一页）链接、一个 Next（下一页）链接，两者当中还有一系列数字。而且绝大多数分页系统都会通过某种方式让当前页码的数字有别于其他链接。

抱着这种想法，我开始尝试为这个论坛的分页系统做一些改变。

首先，我去掉了 Page 标签，把左箭头设置为不可用状态，然后把两个箭头分别放在那串页码链接的两端。这样做是为了让界面在一开始就给人选择：只要愿意，他们可以很快地回退到之前的页面。同时为了一致性，我也修改了箭头的颜色，使之和链接的颜色相配套。见图 10-3。



图　10-3

然后，我将文字的字体加大了一点点，并且改变了当前页码的文字样式，使它从其他数字链接中凸显出来。我取消了该数字的链接属性，把它的颜色改为红色，并且加大了字号。同时也增大了箭头的大小，让界面中每一个元素都能更容易被点击。见图 10-4。



图　10-4

最后，我添加了 Previous 和 Next 的文字链接，把它们放在相对应的箭头旁边。单纯从技术上来说，在大部分情况下这些文字链接都是不必要的，因为箭头足以完成任务。但是，文字链接增大了这两个最为常用的链接的响应区域，使得点击更为容易。记住，界面越简单，交互的瞬间就越美好。见图 10-5。



图　10-5

噢，当然我还把整个分页系统从结果列表页的右上角移到了下方正中。这样它就更容易满足用户的预期，也更容易让用户找到。

在这个案例中，我遵循了标准。设计或者说寻找合适的分页系统变得不再那么困难。诸如 Google 等已经为分页制定了一个非常流行的设计方式，而且也都以非常近似的手段应用这些分页系统。与此同时，这个标准化的解决方案对于哪怕是新手来说也相当好用。很多情况下，与其挑战标准、冒险破坏用户期望，倒不如遵循它们，效果反而会更好。

提供回到搜索结果列表的出口

在我们继续之前，关于搜索结果页面还有一件事情需要讨论一下。

有时候，我从一些网站的搜索结果列表单击进入某个页面，然后在希望回退到之前的结果列表时，却发现该页面中没有这个入口。

当这种情况发生时，用户要么依靠万能的浏览器 Back（后退）按钮，要么直接从这个结果页开始新一轮的搜索。但碰巧我属于坚信用户不应该依赖 Back 按钮的那一类人。网站自身应当始终为用户提供导向，方便他们在站点中来往自如，而不是依靠浏览器工具。

因此，我经常在自己的设计中加入一些东西，从而能够明确地达到这一目的。它们看上去很像 Gmail[①]里面所提供的那样，例如图 10-6。

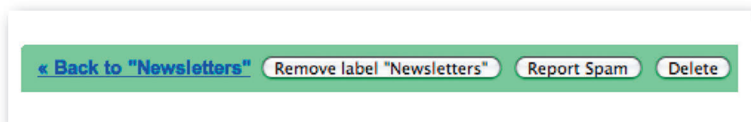


图　10-6

当 Gmail 的用户为邮件设置了某个标签（你能为 Gmail 中的每一封邮件都指派一个标签，类似文件夹那样便于归档整理。这些标签被列在边栏导航的工具条中），并且单击进入该标签下的某封邮件进行阅读时，Gmail 为他们提供了一个 Back to（回到）的链接。

在图 10-6 中，这个链接让我能 Back to“Newsletters”（回到“时事通讯”）。正如你所猜到的，Newsletters（时事通讯）是我创建的一个标

① 众所周知，Gmail 是 Google 提供的免费电子邮箱，网址为 http://www.gmail.com。——译者注

签，里面放的全都是我订阅的该类邮件。如果没有 Back to 链接，我想回到 Newsletters 标签内容列表的唯一方法只能是先在左侧工具栏中找到 Newsletters 一词，然后单击它。

最好的情况下，这让人感觉效率低下。而最坏的情况，很可能我没有想到这个办法，而是碰巧单击了另外什么东西，最后完全迷失了方向。

设计搜索结果页面时，请考虑在其中加上类似上文所提到的东西。例如，Home（首页）→ Search results（搜索结果列表）→ Product page（产品页面）。

我的设计（线框图形态）看上去如图 10-7 所示。

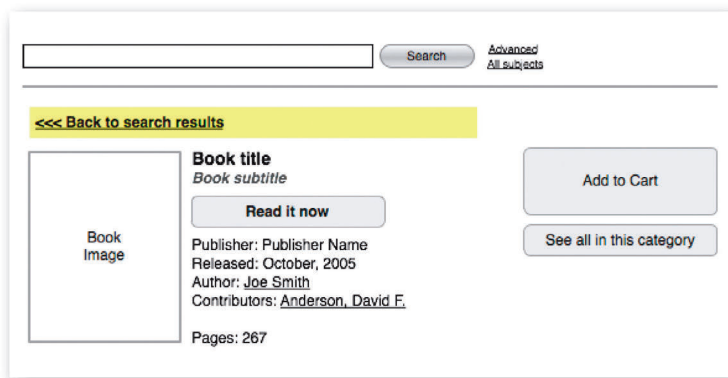


图　10-7

这个背景颜色突出的简单 Back to Search Results（回到搜索结果列表）链接，为用户提供了一个出口。用户有了可依靠的支撑点，从而不需要求助浏览器工具。

好的网站设计应该能够通过我说的"无浏览器自测试"的检验。这种测试抛开浏览器以及站点导航提供的工具栏、收藏夹等一切功能，摒弃任何援助，而只依靠网站本身来引导用户。如果你的站点通过了这一测试，那么今天可以告一段落了，去吃个三明治吧！

Back to 链接和分页系统是非常基础的设计，但如果做得不好，它们就会成为造成用户挫折感的关键因素。当碰到类似这样应用极为广泛的设计时，最好遵循普遍的标准，帮助用户达到他们的目的，而不要迫使他们学习任何新的东西。

第 11 章

精炼你的搜索

在各个网站上，绝大多数人只是依靠那个后面带有“搜索”按钮的输入框来进行搜索。因为这种途径最为简单，也最好辨认。而当需要对他们的搜索加以更多控制的时候，那些网站通常也会提供一个 Advanced Search（高级搜索）的选项以便满足需求。

然而，到底需要有多 Advanced（高级）呢？或者更准确地说，那些高级的选项应该复杂到什么程度，才能让用户找到自己所需要的东西？

很多时候，Advanced Search 就是能塞满整个屏幕的一堆选项，很快就让你难以忍受。比如，图 11-1 中 Google 的 Advanced Search 页面[①]。

图 11-1

① Google 的 Advanced Search 页面现在已经不是作者截屏的那个界面了，已经进行了大幅度的简化。网址为 http://www.google.com/advanced_search?hl=en。——译者注

然而，在大多数情况下，这种高密度的选项其实并没必要。尽管那些更有经验的用户，或者有着非常精确需求的用户可能偶尔会冒险使用这些高级选项，但它们通常可以用一种更为简便的界面展现出来。

下面是我近期做的一个项目，当时正在为某个数字下载公司的产品目录做再设计。网站的快捷搜索，就像其名称那样，确实足够简单，只是一个输入框和一个 Search（搜索）按钮，但是他们的高级搜索就完全变成另一种形态了。不论何时，当用户单击 Advanced Search 链接后，在屏幕上看到的就是如图 11-2 所示的这个界面。

Search content				
Search in:	Text	for:	"interface design"	X
and in:	Subject	for:	"web design"	X
and in:	Publication years	for:	From: 2006 To: 2008	X
and in:	License type	for:	All license types	X
and in:	Document type	for:	All document types	X
and in:	Title	for:	"interface"	X

Search　Start over

图　11-2

这个 Advanced Search 页面并不像 Google 的那样复杂，但是仍然有一些地方可以充分地改进。

该版本的设计者可能认为，如果把所有的选项立刻展现出来，用户就能统观全局，并且只编辑自己需要的那几个选项即可。但是这种方法通常会在用户头上形成那种讨厌的卡通思维泡泡：

“我需要填完所有这些玩意吗？如果我只填写了一部分，会发生什么？”

随着更为仔细的检查，用户可能会注意到那些用来删除表单中某行选项的 X 图标。而这些又会触发另一个思维泡泡：

“这些图标是说我必须从表单中移除不需要的选项才能正常工作吗？那我移除的太多了怎么办？还有，如果我改变主意了，想再加回来，该怎样做？我是否必须单击 Start Over（重新开始），然后，再全部从头再来？”

当然，还有：

“如果我的这次搜索是这样设置的，那以后怎样才能够再次使用同样的设置？”

要想清除所有这些问题泡泡，只需要把这个表单整理一下即可。

11.1　高级搜索也简洁

首先，我决定保留默认的 Quick Search（快捷搜索）选项，因为它放在那里非常合适，非常有用，也非常便于使用。大多数用户想要开始搜索时都会从这里开始。而且它表现的方式非常标准，因此我找不到什么理由来改变它。见图 11-3。

aviaton　Search　Advanced　All subjects

图　11-3

为了在新的设计方案中稍微提点速，同时让表单更有效、更容易理解，我希望能避免像现在这样把用户送到一个新页面下，然后让他们填写一个复杂的表单。

我在这里使用了“行内扩展”的方式。新的元素将会凭空“掉落”到网页中，然后把其他的内容往页面下方推开。推开后，就如图 11-4 所示。

aviaton　Search　Advanced　All subjects

Search content

Search in:	Text	for:	"interface design"	X
and in:	Subject	for:	"web design"	X
and in:	Publication years	for:	From: 2006 To: 2008	X
and in:	License type	for:	All license types	X
and in:	Document type	for:	All document types	X
and in:	Title	for:	"interface"	X

Search　Start over

图　11-4

通过这种方式，用户单击 Advanced 链接后新的表单将会如戏法般地出现在当前页面，而不需要等待载入新的网页。

足够简单。但真正的问题并不是页面载入的时间，而是那个丑陋的表单。

11.1.1 递进显示

绝大多数人都不愿费事去逐行逐项地编辑这个表单。而在可能会填写这个表单的少数人当中，也只有更少的人会每次都这么干。通常来说，那些确实需要高级搜索选项的人其实只需要很少的选项。

换句话说，人类的行为模式告诉我们，没有必要一次就把所有的选项都摆出来。我们真正需要的是一个简单而又能够广泛应用的解决办法，最好能只在必要时才增加选项。

这种时候 Progressive disclosure（递进显示）就是我们要找的完美解决方案。

简单来说，所谓的“**递进显示**”[①]就是一种设计方法，它能根据实际需要（即依靠用户的输入）而逐步显示控件，而不是默认把它们一股脑全堆在眼前。

对话框通常能体现出递进显示的概念。比如说，单击 Advanced Options（高级选项）链接可能会弹出对话框，里面能够展示更多的选项。这些选项在默认情况下是隐藏的，而只有当用户确实单击了某个链接让它们出现时，它们才会出现。

为了让这个想法应用到 Advanced Search 表单中，我用 1 行单独的控件替换了原有的 6 行控件，并且在后面加上了一个增加控件的按钮，如图 11-5 所示。

通过这种方式，用户能够在需要的时候，只需一次增加一行。他们绝不会因为突然出现一大堆选项而难以忍受。

① 本书作者有一篇专门讨论“递进显示”方法的文章，2007 年 9 月发表于 Vitamin 网站，题目是“Read more……关于递进显示方法”。网址为 http://www.thinkvitamin.com/ features/design/read-more-about-progressive-disclosure。——译者注

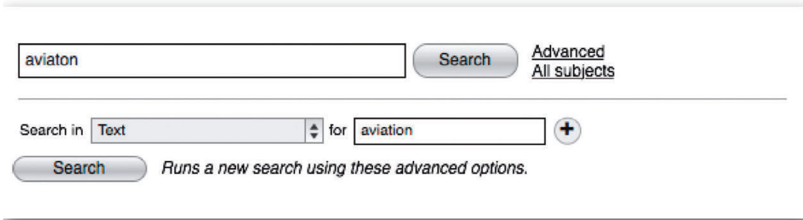


图　11-5

同时我也去掉了 Start Over 按钮。为什么？想一想，你最后一次目睹有人点击这种选项是在什么时候？绝大部分时候用户不想全部从头来过。他们只希望修改部分内容，这样就行了。所以，送它上路吧。

（顺便说一句，对于其他类型的网页表单来说，这种做法也是正确的。我们常常看到的 Clear Form（重置）选项其实一点用都没有。）

11.1.2　引人入胜的交互

我留下的 1 行是用户已经在 Quick Search 中输入过的内容。在图 11-6 中，选项表示搜索所有文本中带有 aviation（航空学）关键词的内容。但很明显，单击了 Advanced Search 的用户希望在他已经做完的基础上进一步地精炼搜索。为了改进这一点，我在它下方又加上了一行有待输入的选项。

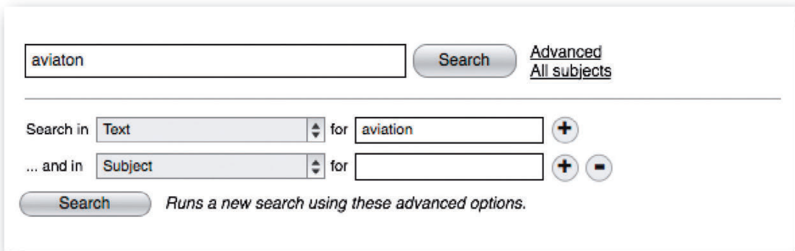


图　11-6

这种方式告诉了用户如何为搜索增加条件。而且在理论上，它有可能会诱使他们自己去增加选项。

唯一的问题是那个 Quick Search 仍然是激活的，而同时存在两个外观彼此不同的 Search 按钮可能会混淆视听。

最简单的解决办法就是让 Quick Search 失效（图 11-7 顶部的 Search 按钮）。

图 11-7

而同时我把之前的 Advanced 链接改成了 Hide advanced options（隐藏高级选项）。这样用户就能知道如何取消交互，然后回到 Quick Search 中去。

“递进显示”是一个简单的概念，而且它能以各种方式适用于绝大部分的 Web 应用。在这里我们用它来简化高级搜索表单，但在其他地方它还能用来隐藏各种高级功能，甚至还能为那些少见的配置选项提供一个容易辨识的入口。

此外，如果你的老板有一大堆差劲的主意，而又要求你必须实现出来，那么你也可以运用“递进显示”的方法来隐藏其中的内容。

只要别告诉他是我说的就行。

第四部分

深　　入

假设我们的网站已经为用户提供了充分的导向，那么现在是帮助他们深入去学习更多东西的时候了。比如注册账号，以及开始真正的应用。

每一处交互其实都是一个为用户创造流畅、愉悦瞬间的机会。

我们可以提供 Screencast 或者其他界面友好的演示视频来帮助用户更容易地了解某项服务或某个应用程序（正如第 8 章所描述的那样）。我们也同样需要把这种交互行为本身设计得让用户再一次地感到自己有成效、聪慧而且备受尊重。然而，互联网上关于视频方面却又没有一个能让人依葫芦画瓢的公认标准。考虑到这些，本部分第一个故事讲述了如何设计一个能在各种情况下都通用的视频播放器。

一旦我们说服某位用户注册之后，他就会发现自己面对着一个填写资料的过程。而网站的表单又总是充满着各种潜藏的错误，哪怕连最简单的表单也可能会导致混淆。绝大多数表单都只是使用默认的 HTML 控件，然后就有失考虑地扔到界面中去。成千上万的表单几乎都一模一样，它们没有反馈机制，没有即时校验，也毫不考虑是否能支持用户的思维模式。

尽管每开始设计一个新的表单时，我总是认为只是再寻常不过的又一个表单而已，但实际上在每次结束时我总会很享受。因为即使是一个简单的注册表单也同样会有需求和约束，以及各种挑战。每一个表单在设计出来后都会和以往的有些不同。

我留意过很多人使用过很多表单，而且了解到什么样的表单会让人觉得沉闷无聊，又是什么样的表单让人用起来不会感到丝毫痛苦，甚至会成为愉悦的享受。

正因为如此，我会向你们讲述几个关于设计表单的故事，从最简单的说到最复杂的。这些故事针对的内容包括表单排版、即时校验、分步拆分长表单、标签、找回密码，甚至还包括表单间相互控制条件。

如果运气好，你们将会开始用一种全新的眼光来看待表单，并且意识到它们对用户体验将造成难以置信的影响。

不过首先，让我们还是看一眼如何改进视频播放器来帮助用户在互联网上操作视频文件。

第 12 章
视频播放器的标准化

我们对于通过互联网操作视频文件早已司空见惯。开发者只需要创建一个用于播放控制视频文件的界面，然后就能把各种视频注入到播放器里面去。但出于某些原因，各种在线视频播放器相互之间和世界上其他万物一样千差万别。

YouTube、Yahoo、CNN、NYTimes 等网站都有很不错的播放器，其设计几乎轻易就能成为业界标准。但是大多数开发者似乎并不爱从主流站点吸取经验，也不愿自己的网站受其影响。当然，仍有一些主流网站似乎正在相互学习，逐步趋近于找到一种更为标准的方法来规范视频操作。而这正是我们现在所要探讨的地方。我们将看看网上那些最经常使用的播放器是如何设计的，并以此尝试创造出一种博采众家之长、在绝大部分情况下都能顺利运作的播放器。

尽管我曾经是一个 Adobe 的 Flash 开发人员，但并没有什么客户请我设计视频播放器。所以我纯粹只是为了好玩而设计了一款。我相信它在各种网站和应用程序中都能完美运作。

通过这个例子我希望告诉你们，对那些已经重复百万遍的事物加以研究和改进是件多么简单的事情。

12.1　拨开 VCR 设计的迷雾

要想改进视频播放器，你必须先透彻地了解现有播放器的设计。因此让我们来看看一些最著名的播放器。

先看 NYTimes.com[①]的播放器，如图 12-1 所示。

① 网址为 http://video.on.nytimes.com/。——译者注

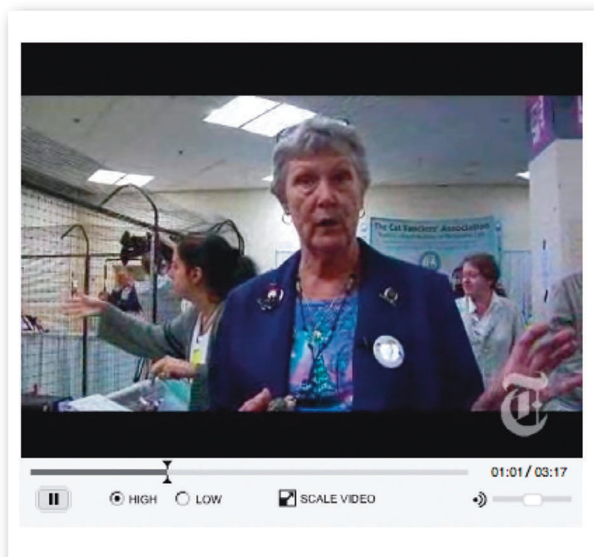


图 12-1

这是一个相当有代表性的视频播放器，显示了视频的长度、当前已播放时间、音量控件、播放 / 暂停按钮，以及一个进度条。但是，有代表性并不意味着没有问题。

首先，它的滑块按钮（在视频时间轴上用于拖动的滑柄）看上去极其难以使用。它又窄又小，简直到了可笑的地步。而实际上该滑柄的响应范围其实还蛮大的，但视觉上感觉实在太难点击。我相信这里肯定会在一开始就给用户留下一种难以使用的印象。

其次，音量控件的滑块太小。上年纪的用户，尤其是那些患了关节炎或手部活动不便的用户，在移动那些需要精准度的控件方面可能会有麻烦。

CNN.com[①]的视频播放器也有类似的问题，见图 12-2。

音量滑块太小，很难操作。文本也太小，凡有轻微视力问题的用户都很难阅读。

尽管这样，CNN.com 播放器还是有很多东西让我相当喜欢。

① 网址为 http://www.cnn.com/video/。——译者注

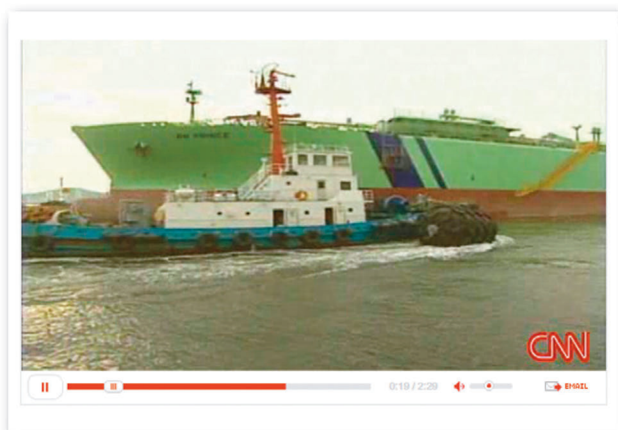


图　12-2

首先，CNN 抵制住了自动播放视频的冲动。这意味着在页面一开始载入的时候占用的带宽更少，同时也意味着那些不想看视频的用户无需被迫手动停止视频，从而能够安静地阅读页面。

其次，页面载入后，播放器在视频框的正中心提供了一个大大的播放按钮。它提示用户当前显示的图像并不是静态图片，而是来自视频（另一个提示是播放器控件的可视化设置）。同时如果用户选择观看视频的话，它也提供了一种简单、明显的方式让他们点击播放。一个按钮做了两件事。如图 12-3 所示。



图　12-3

YouTube 的播放器同样也提供了一个大播放按钮（图 12-4 中显示的播放器是我们经常在 YouTube 之外其他网站上看到的嵌入式版本）[①]。



图 12-4

然而，YouTube 播放器也有一些问题。

首先，尽管单击视频框正中的 Play（播放）按钮可以直接播放页面中的视频，但如果暂停后再次单击视频框时，却突然会弹出一个新窗口链接到 YouTube 网站中去。这一点打破了用户的预期，因为他们认为单击视频框自然会开始播放。

其次，播放时间那里仅仅只显示了剩余时间，而不是视频的整个时间[②]。见图 12-5。

用户也许能够习惯 YouTube 这种与其他网站不同的时间显示方式，但并没有什么理由一定要有这种不同。

---

① 图 12-4 中的视频名为 *The Paradox of Choice (Barry Schwartz)*。网址为 http://www.youtube. com/ watch?v= VO6XEQIsCoM。——译者注

② 现在 YouTube 播放器在该问题上已有所改进。——译者注



图 12-5

而且，很明显，文字和控件都非常小。

最后，视频的标题并不在嵌入播放器里。所以当有人想在自己网站或博客上嵌入某个 YouTube 视频时，他们必须还得加上文字来说明这个视频是什么。这通常也意味着他们需要来回切换浏览器窗口，逐字地确认视频名称并写下来。

YouTube 成功的秘诀不正是那种可以随处嵌入的能力吗？正是由于这种功能，你才能在自己的网站上和自己的观众分享。那么他们为什么要把视频的标题排除在外呢？把它放在播放器里面难道不会让上述这个过程更为简便吗？

总而言之，这些播放器中的每一个都做对了一些事，但还有些事又做得不是很漂亮。而我们就是要总结并学习所有它们干得不错的地方。

12.1.1　且慢——还有做得更好的

我还想谈谈另一款播放器，它来自于一个并不是很流行的网站，但却让我非常欣赏。这款播放器来自于 TED.com，传奇般的 TED Talks[①]的视频版本网站，如图 12-6 所示。

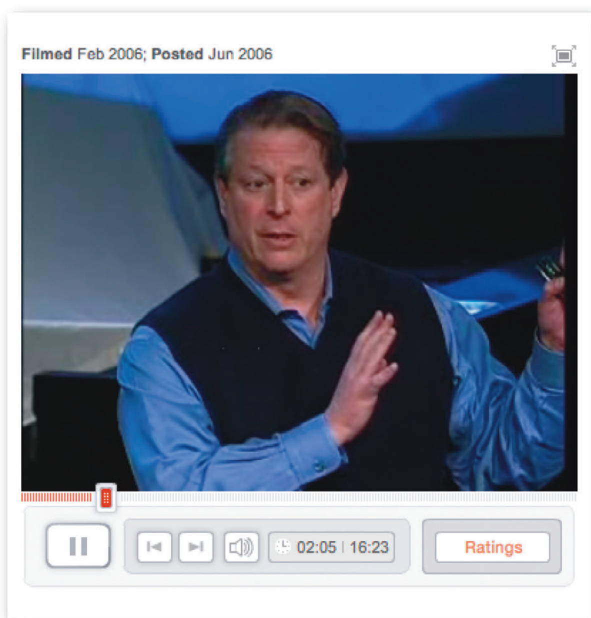


图　12-6

这里，Al Gore 正在演讲我们如何才能协助减缓全球气候危机的问题。

控件的尺寸相当合适，标准的字体大小让文本更易阅读，而左上角还提供了一些关于视频的附加信息，例如，录制和发布的具体时间。

① TED Talks 来自于 TED（Technology、Entertainment 和 Design3 个单词的首字母）。自 1984 年开始，TED 以定期会议的形式聚集技术、娱乐和设计界的人士分享并传播创意。TED.com 是 TED 于 2006 年发布的免费视频网站，汇集了 200 余个精彩的演讲视频。——译者注

(TED 不需要在播放器中包含视频的标题，因为 TED 视频不能直接嵌入到其他网页，除非你找到了它们的 YouTube 版本。这颇具讽刺意味，不是吗？)

由于视频的长度超过了 16 分钟，它被很贴心地划分成数个章节，而用户可以通过播放器的菜单系统来跳转到其中的任意部分。见图 12-7。



图　12-7

相当不错，对吧？你所要做的只是把鼠标移动到视频框上，这个菜单系统就会自动出现。

TED 播放器我发现的唯一的大问题就是它没有提供必要的提示，让用户明白只要鼠标悬停在视频框上就会出现菜单系统，从而能够在章节中提供导航。用户只有在偶然把鼠标（也许是其他什么原因）移动到视频上才会知道有这一功能。

另一个小问题是时钟图标过小，它对于任何视力低于 20/20（正常视力标准）的用户都很可能会毫无用处。而没有时钟图标的话，人们只能假设这些数字是和时间相关的。在没有明确标记的情况下，用户必须去思考数字代表什么，而人类似乎又总是在逃避思考。

12.1.2　博采众长

总的来看，TED.com 的这个可以说是网上最好的一个播放器了。因此它是一个很好的起点，让我们开始设计一款新的、能在绝大多数环境中通用的播放器。

为了让它更通俗有效，我需要调整几处我认为不太合适的地方，然后把 Ratings（评级）按钮去掉，换成章节导航的菜单系统入口。

简而言之，我希望做成如图 12-8 所示的样子。



图　12-8

这个版本中，我在左上角加上了视频标题，用 Time（时间）标记替换了之前小的时钟图标，然后把 Ratings 按钮变成了 Chapters（章）按钮。

现在，单击 Chapters 按钮将会打开菜单系统，以便选择视频中的任一章节。见图 12-9。

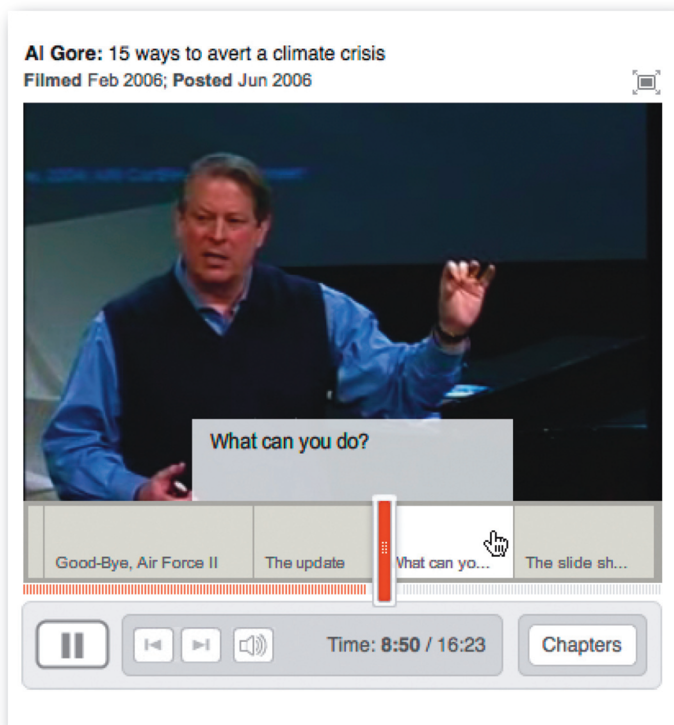


图　12-9

但在你开始考虑划分视频章节所需要付出的额外工作之前，我必须指出，并不是每一个视频都有或者需要划分章节。

所以我为播放器又添加了一条规则：如果没有章节，就去掉 Chapters 按钮。如图 12-10 所示。

现在，这款播放器提供了所需要的一切核心功能、相对更易用的控件、清晰可辨的文本，以及视频的相关信息（长度、已播放时间、标题以及录制和发布的日期）。

最后，我决定把它设置为默认停止状态，避免页面载入时讨厌的自动播放问题。然后在视频框的中心加上一个又大又漂亮的播放按钮，只要视频停止时它就会出现。



图 12-10

搞定！

嘿——如果什么时候需要为你的网站设计一款视频播放器，给我电话。我已经为你准备好了相当棒的一款。

噢，现在就有电话来了。

第 13 章
巩固你的表单布局

很幸运，我们在网上遇到的绝大多数表单都不算很长。它们仅仅包含一些表单域，也许还有 1 到 2 个下拉菜单，再加上一个按钮。而且它们能帮助我们完成想做的事，让我们能在某个新的网站或应用上注册、购买商品或者联系某家公司。

但你想过没有，就连一个短短的表单这样的东西，为什么在不同的网站里看上去差异会如此之大呢？

有些表单布局成两列，标签呈左对齐，还有一些则是呈右对齐。另外一些表单只有一列，而标签位于表单域的上方居左对齐。Submit（提交）和 Cancel（取消）等按钮要么在表单左侧要么在右侧，有时候 Submit 在 Cancel 左边，有时候又在它右边。还有标签……噢，别让我说标签，根本就说不完。

很无奈。这么简单的事情都会 千差万别。

很多情况下，这些千姿百态的设计只是单纯客观条件下的产物。也许是因为设计师认为在页面上标签右对齐会更美观，应该为了配合页面来设计表单而不是围绕表单来设计页面。或者是因为开发人员总习惯于以自己网站上的表单布局方式来为所有其他网站布局。还有更甚者，也许只是某个人直接从另一个网站复制粘贴了代码，而又懒得修改。

不管出于什么理由，但其实哪怕是没有经过专门设计的表单，同样也可能创建美好瞬间，从而为用户整体的操作体验做出贡献。不过很多表单都错过了这个机会。

你也许没有意识到这一点。事实上，每一种布局都会对用户使用表单（以及你的网站）产生积极或者消极的影响。而在正确的时间选用正确的布局，才是提升用户操作体验的关键，而且这也很容易做到。

13.1　设计出行云流水般的表单

表单需要呈现出延续的流动性，而表单的布局则极大地影响到用户是否能够成功地填写表单。例如下面这个为某在线社区网站设计的注册表单。

这个表单只需要收集用户的真实姓名、电子邮件地址、他们希望的用户名和密码，以及必要的信用卡信息。

我先考虑了两列的布局，标签居左对齐。如图 13-1 所示。

图　13-1

这使得表单很好看，也很简短，看上去所有内容都能在页面上以 Above the fold[1]的形式呈现（位于默认的可视区域，无需滚动条），然而这种布局方式较之其他布局却更容易引发错误。因为在第一列和第二列，也就

① Above the fold 这一术语尚无准确的中文翻译，该词起源于报纸的排版原则。Fold 原意为报纸被折叠起来的部分，排版人员应力图避免重要文章（如头条新闻）的标题、内容或图片横跨折叠处从而造成浏览者阅读不便。这一概念在 Web 领域中则取决于用户显示器分辨率的 Y 轴大小。——译者注

是字段标签（例如电子邮件）和与其对应的输入域间的映射关系很不明显。

呈左右两列的布局同时也降低了用户往下浏览的速度。如果是那种需要仔细阅读的表单，可能这样会很好，但是也同样容易引发错误。而作为标准的注册表单来说，我们没有必要去减缓用户往下浏览的速度。于是我先考虑了右对齐版本，如图 13-2 所示。

图　13-2

标签居右对齐时，用户不会像左对齐时那么容易犯错，因为标签和输入域之间更为接近。然而，右对齐的标签在页面的左侧形成了一道难看的、锯齿状的边缘，这种平衡性的失调完全没有吸引力。另外，不管标签如何摆放，两列的布局仍然会减缓用户往下浏览的速度。

所以我决定直接把布局变成一列来显示，而标签则放在每个输入域的上方，居左对齐。见图 13-3。

Register

First name:

Last name:

Email address:

e.g. me@mydomain.com

New password:

Confirm new password:

Name on card:

Card #:

Expiration:　Security code:

01　07

OK　Cancel

图　13-3

这种布局让用户只用花最少的时间就能将标签和输入域对应起来，因此浏览速度比两列布局要快不少。用户只需在同一个方向移动视线就能看到标签和输入域，而不用先阅读左列的标签然后再瞥一眼右侧的输入域。这样改善了的分组，使单列布局同样也能减少错误的发生。

我设计的大部分表单都使用了单列布局，标签位于输入域的上方居左对齐。我时常考虑表单的布局会怎样影响到用户与之交互的瞬间，而最后总是选择了这种布局方式。只在很少的情况下我才会有意地在用户填写表单时减缓他们往下浏览的速度，而这种时候我通常更多依靠的是优秀的校验技术，而不是真去给他们造成不便。我们会在第 15 章再讨论这个问题。

13.2　让“确定 / 取消”更完美

哪怕最简单的表单也有个地方可能会诱导用户犯错误。那就是在它们最下方出现的讨厌的 OK/Cancel（确定 / 取消）按钮。

网络上 OK 和 Cancel 按钮无处不在。当然，它们并不总是标记为 OK 和 Cancel，也不一定每次都是同样的用途，但是我们每个人都曾遇到过上百万次了。这两个按钮的组合简直是横扫天下，也出于一个很不错的理由：在 Web 应用中几乎你的任何行为都能够取消。

我们经常在那种可以往返的交互中看到它们。用户从第一页开始，然后在第二页完成交互，如果单击 Cancel 则回到第一页。它们也会出现在同一个页面内的即时交互中，例如在某些在线工具网站上修改个性设置。

这对按钮最有代表性的就是陈列为两个并列的按钮，而且采用标准的浏览器控件外观。这种设计可悲地持续了很多年，而且应用于无数个应用程序中。

我们这个表单也以这种最具代表性的按钮套装开始。如图 13-4 所示。

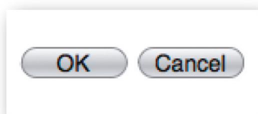


图　13-4

本例中用户交互的目标是注册，就这么简单。但是任何一个设计问题都可能有无数种方式来解决，而即使是在这个最简单的例子里，也同样需要做几个决定。

第一要决定按钮的位置。居左对齐还是居右对齐对于可用性来说并没有很大的影响，但是因为下述理由我一般仍然坚持把它们放在左边。

首先，左对齐使所有元素都依靠左边缘顺势而下，从表单顶部到底部形成了一条好看的直线。其次这也是一种美学上的选择，因为看上去它为表单建立了一个视觉上的支撑点。在完成每一处表单域后，你会继续规则性地往下移动。把动作按钮放在表单的尾部看上去就好像把一切都钉牢固定下来。当然这也许只是我的个人感觉。

第二要决定按钮上的文字。当然，在绝大部分情况下，使用标准的 OK 和 Cancel 最为省事，因为你不用作任何考虑，就这么标记也完全可以过关。但是这两个术语真的能带来帮助吗？

OK 在大多数时间里都能正常工作，但是意义更明显的按钮标记能更好地帮助用户建立对点击结果的期望。

这个例子中单击 OK 会发生什么呢？它将把这位浏览者注册为网站的一个新用户。但在这个网站的其他地方，OK 也许意味着保存某项设置，或者上传某个文件。

单击 OK 不能使用户建立起一个清晰的期望。幸运的是，标记按钮并不是一件很难的事情。我所要做的只是把按钮上的文字改成 Register Now（现在注册）而已，相当简单，见图 13-5。

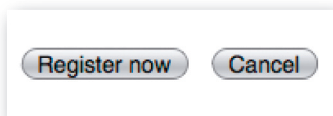


图　13-5

13.2.1　首选处理方式和第二处理方式

这一部分要考虑的最后一件事，可能对于设计师来说常常是毋庸置疑的。那就是我们经常赋予这两个按钮同等的重要性，而它们却会引发完全不同的结果。一个用于用户注册，另一个则取消整个操作（这个例子中将把用户带回到首页），但两个按钮在外观上却一模一样。

我能理解这种传统形成的原因。两个选项分别对应于两个按钮。然而，重要的并不是选项的数量，而是哪一个选项才是最可能的选择。

对于决定在网站注册并且填写表单的用户来说，他最有可能选择去单击 OK。这被称为**首选处理方式**（Primary Action）。而取消整个注册过程则是不太可能的选择，它被称为**第二处理方式**（Secondary Action）。

然而，如果两个按钮都被赋予了同等的视觉重要性，那么用户则必须阅读按钮上的标记，而且两个按钮都得阅读以便决定应该点击哪一个。

费茨法则（Fitts’ Law）[①]指出，点击某个目标所花费的时间可以通过一个函数来求值，它与“鼠标与目标的距离”和“目标尺寸”密切相关。根据这一法则，我们最好的解决办法就是从第二处理方式，也就是 Cancel 按钮上减弱一些注意力。于是我把它变成了一个文字链接，如图 13-6 所示。

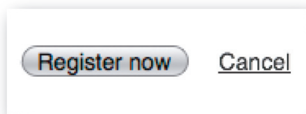


图 13-6

这样一来，Register Now 按钮变得更为突出、也易于点击了，同时 Cancel 链接也不再那么明显，相比之下不那么容易点击到了。

这两个按钮是不是变得很难看到？没有。但对用户来说，现在点击失误倒是不再那么容易了。用户的视线会被牵引到大的按钮上去，而且它很容易让人下意识地认识到大的按钮很可能就是首选的处理方式。如果那正是用户想要的，他们几乎都不需要去阅读按钮上的文字。如果用户想要的是第二种选择，他们也能轻易推断出不突出的那个选项就是 Cancel 链接。（然而有点可笑的是，在两个按钮间创造视觉上的差异将使得它们更易于阅读，因此尽管用户能够在不需仔细阅读标记的情况下就能推断出它们的含义，他们反而更想去阅读这些文字。）

顺便说一句，这也是我喜欢让按钮居左对齐的另一个理由。

如图 13-7 所示，把大 HTML 按钮与其上的表单域对齐，看上去明显比让 Cancel 链接与上面居右对齐要好得多。

另外，为什么要把按钮放在左边而文字链接放在右边呢？因为那些使用 Tab 键切换输入域的用户将先到达 Register Now 按钮。这再一次验证了我们应将焦点集中在可能性更大的选项上，同时降低用户在可能性较小的选项上的注意力。

① 费茨法则是人机交互领域里一个非常重要的法则，10 年来得到了广泛的应用。其最基本的观点就是当一个人用鼠标移动指针时，屏幕上目标的某些特征会使得点击变得轻松或者困难。目标离得越远，到达就越费劲。目标越小，就越难点中。表达这一概念的公式为：Time = a + b log2 (D/S + 1)。其中，Time 代表“从鼠标到达目标的时间”；D 代表“鼠标达到目标的距离”；S 代表“目标的尺寸”。——译者注

图　13-7

13.2.2　这很重要

现在，千万不要再弄错了。对于任何一种布局来说，完成表单花费的时间多几秒或者少几秒，并不会对用户体验造成很大的影响。归根结底，最重要的不是在这一过程中抠出那珍贵的数秒钟，而在于完成表单所带来的价值是否能超过填写表单时给用户带来的麻烦和沉闷。

在一开始劝说用户填写表单是整个注册过程中最困难的部分。而一旦你做到了这一点，就应当像对待真正用户那样尽可能地善待他们。在表单的设计中考虑每一种方案间细微的区别，也就是真正注意到每一处细节，这将会为用户创造另一个平滑顺畅的瞬间。而每一个瞬间都会对整体的用户体验做出贡献。

如果你正在花费所有的时间和精力来创建某个 Web 应用，为什么要把自己网站的表单设计扔给运气或者客观环境去评判呢？主动地去考虑和研究，你必定会收获更多愉快的用户。

第 14 章 驾驭 Wizard 向导

我曾参与过一个广告宣传运作管理系统的设计工作。其中创建宣传活动的流程需要一系列用户以特定顺序填写表单。用户首先需要创建一个活动，包括指定活动名称、开始和结束日期，以及其他相关信息，然后，才能在该活动中创建广告并设置经费额度等。

桌面应用和 Web 应用一样，它们经常都会包含一些必须以线性方式逐步完成的操作流程。而 Wizard 向导通常被用来解决类似的问题。它是一种多步骤的交互方式，用于引导用户一步一步地通过那些有固定顺序的处理过程，并且通常被用来解决类似的问题。

目前，基本上任何古老的 Wizard 向导设计都能帮助用户完成这种过程。但是有良好的指示器为用户预期表单复杂度提供线索，更能够增进用户的信心，并且能为交互创建一种清晰、易于理解的瞬间。

具体地说，我们可以通过提供一系列“路标”来为用户建立清晰的预期。

14.1　建立清晰的预期

首先让我们考虑当 Wizard 向导缺乏必要的指示器时会发生什么。

图 14-1 的界面中唯一能告诉你这步操作并不是整个交互最后一步的元素是 Next（下一步）按钮。这里没有任何东西能让你知道创建整个宣传活动总共需要多少个步骤。

图 14-1

在这个设计中，你填写完页面中的表单然后单击 Next。下一个步骤会载入一个新页面，然后填写表单，再次单击 Next，等在它后面的又是另一个表单。你在事先肯定没有预料到这一点，所以每当看到另一个 Next 按钮时，你都会被迫去猜想为了完成这个创建过程到底还需要多少个步骤。

我到哪一步了？还有多少步？每一步有多长？填完这些是不是还需要很长时间？我是否应该取消今天的晚餐计划？

这对用户建立自信心真没什么帮助。

你也能很轻易地去除用户的挫败感，只需要为他们建立期望，让他们预先知道总共有多少个步骤。

同时我们也应该告诉他们去过哪里，现在在哪里，将来还要到哪里。

这很简单，有点像为每个步骤加上一个描述标题栏。图 14-2 里面的标题栏指出共有 4 个步骤，用户此刻正处于第一步，而第一步是关于 Create a campaign（创建一个活动）。

仅这一点就是个很大的进步，不过它还可以变得更有效果。

将所有步骤的名称都展现给用户将会更有帮助。通过这种方式用户可以随时瞥一眼自己正在哪个阶段，然后决定是否继续。为了做到这一点，我尝试了一种标签页界面的变体。

图 14-2

在图 14-3 这个版本中，用户能更容易地看到共有 4 个步骤，也更容易明了他们即将需要什么类型的信息。

图 14-3

当前步骤是绿色的，而随后的步骤不可点击，因此用户不能随意弄错顺序。每当完成一个步骤后，相关的信息将会被记录，在下一步中系统会根据之前的选择进行验证。

设定范围界限

然而，就如我们都知道的网页上的垂直空间是无限的。页面有多长并不重要，滚动条会自动适应并让我们来回查看它。正因为这一点，在现有的设计中我们无从知道每个步骤到底有多长。

“告诉大家，第二步里面包含了 26 个多项选择、1 个字谜，以及 1 个问答题。”

我希望能为用户建立一种比上述更为清晰的预期。我希望用户只要瞥一眼表单，就能推断出完成整个过程不会超过 5 分钟。

为了实现这一点，我需要体现出 Wizard 向导里的每个步骤真的都非常短。唯一能做到的方式就是为这个区域设定一个固定的高度。

换句话说，我需要一个可折叠面板控件来取代现有的标签页。如图 14-4 所示。

图　14-4

可折叠面板控件是一种和手风琴非常相似的界面[①]。它包含数个面板，依据用户的输入而展开和收拢。图 14-4 中，第一个面板是**展开**的（可见或打开），包含了 Create a campaign 表单。而第二、三、四个面板则都是**收拢**的（隐藏或关闭），其中的每一个面板都不能点击，以此来预防用户以错误的顺序访问某个过程。

这个设计中，用户填写完当前表单并单击 Next 后，第一个面板将滑动收拢，而第二个面板随之滑动展开。用户重复这一过程直到第四步结束。

因为整个界面的高度是固定的，用户很容易就能推断出在这个过程中每一个步骤所包含的表单都会和第一步里面一样简短，甚至更短。这样他们就能建立一种期望认为自己能在很短时间内就可以完成整个表单。

和之前的设计一样，过程中每个步骤都有一个 Next 按钮。而第四步是最后一步，所以我用了一个动宾词组来代替 Next，见图 14-5。

图 14-5

① 可折叠面板控件和手风琴的英文一样，都是 Accordion。——译者注

这样很清楚地表明用户正在保存所有的信息，他的活动正在被创建。

最后一步，我把每个已访问过的面板标题改成了链接形式，以此告诉用户可以返回之前那些面板去修改已经输入的信息。当然，有些改动可能会影响到后面的面板。如果发生这种情况，你可以在用户每次做出这类改动时显示一条消息说明这一影响，并且要求他们重新进行后续的设置。

通过显示步骤数并且帮助用户推断每个步骤的长度，我们为用户建立了清晰的期望。他们能够知道自己正在做什么，还能预估完成整个过程所需的努力，同时还可以获得一个愉快的，或者至少是相对轻松的瞬间。

第 15 章
最后的一英里：即时校验

如果不讨论长度或复杂度，表单设计的最关键问题之一就是用户可能会在填写过程中犯各种错误。

犯错误时我们会责怪自己，用户也一样。绝大部分表单都只有很简短的标签，同时也缺少提示性的默认值，而最普遍的莫过于没有任何预防及侦测错误的方法。即使我们绝大部分人可能都认为自己相当聪明，可一旦错误发生，我们却总认为是自己的过失，而不是指责该程序设计得有问题。

自我指责不可能创造美好瞬间。它和我们所希望的用户使用感受（和我一起念：有成效、聪慧而且备受尊重）完全相反。

因此，要想改善产品，我们最需要做的事情就是无论何时都尽力杜绝错误的出现。而正因为用户在表单里到处都有可能犯错误，它倒是一个很好的开始。

很多表单使用了一种简单又明显的方式提示错误：JavaScript 的警告对话框。见图 15-1，它出自 GoDaddy.com[①]的验证程序。

由于警告对话框非常容易创建，你只需要一行 JavaScript 代码，因此它们现在已变成了互联网上报告错误的标准方式。

但现在的问题是人们已经习惯了这种告警方式。他们不再阅读上面的文字，而只是被动地扫一眼，然后便单击 OK 按钮。如果用户根本不去读，或者说他们单击 OK 只是出于习惯而不是因为理解，那么显示这种警告对话框几乎一点用处都没有。

---

① GoDaddy 是世界第一大域名注册商，进军主机领域后发展迅速。目前放置在 GoDaddy 上的网站数量已居世界首位。GoDaddy 同时提供 Linux 主机、Windows 主机、VPS 以及独立主机全线主机产品，各种需求的客户在这里都可以找到适合自己的产品。网址为 http://www.godaddy.com。——译者注

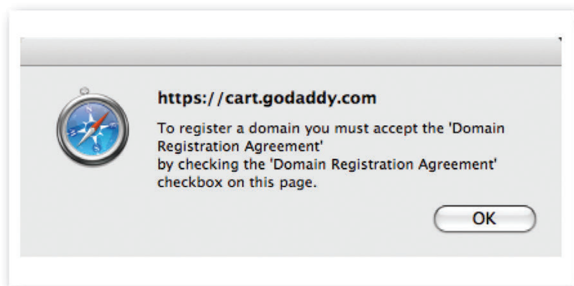


图　15-1

同时，警告对话框是一种**应用程序模式（application-modal）**[①]，也就是说在警告被处理之前，浏览器功能将被强制失效。然而实际上，真正严重到需要用户必须马上处理的错误并不多，所以动不动就锁住他们的浏览器其实是相当粗鲁的行为。而现在却是几乎每一个错误都要变成警告对话框来打断用户的正常工作。没有别的原因，只是因为警告对话框很容易创建。

为了控制住你自己，记住下面这句话：

警告对话框用得越多，就越没有效果。

它们应该用于应付紧急事件，例如当网站着火时用来敲碎放灭火器的玻璃。

对于错误信息以及校验来说，它们几乎完全可以用不那么粗暴的方式来展现，从而创建一个更为友好的瞬间。

在为 UpDown-Repeat 网站[②]的校验表单进行设计时，Kris Hadlock[③]（来自 Miskeeto[④]的一位顾问）和我致力于改善这一局面。我们设计出了一套

---

① 应用程序模式对话框将阻止用户和该应用程序发生其他任何交互，但用户可以切换到其他应用程序。读者可以通过以下链接查阅 Apple 的人机界面指南（Apple Human Interface Guidelines）中关于 application-modal 的详细说明 http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGWindows/chapter_18_section_7.html。——译者注

② UpDown-Repeat 是一个设计界的工作坊（Workshop）网站，用户可以在线发起或参加各类型工作坊。但该网站现已不再经营。——译者注

③ Kris Hadlock 是一名 Web 设计师和程序员，同时也是一位作者。他成立了 Studio Sedition（暴动工作室），网址为 http://www.studiosedition.com/。——译者注

④ Miskeeto 是一个世界级的网页设计、开发及战略研究的团队，领导者是 Robert Hoekman, Jr.（其实也就是本书作者）。他们聘请的顾问都是来自世界各地的专家及有杰出贡献者。该团队的网站为 http://www.miskeeto.com/。——译者注

解决方案，在事先便尽可能多地减少错误，如果仍有漏网之鱼则礼貌地通报给用户，竭尽所能让该表单不再给用户造成麻烦。

15.1　要纠错，也要赞扬

UpDown-Repeat 网站中，用户如果想出席某个研讨会或工作坊，需要输入本人的身份及账户信息。而在这个填写过程中几乎每一个表单域都有可能引发错误。

可能用户输入的电子邮件地址无效，或者信用卡号里漏掉了一个数字，或者遗漏了某个必填项，或者其他诸如此类的错误。

那么我们怎样才能制止这种令人抓狂的局面呢？

可以学习已有的不错实例，比如 Eventful 网站的注册表单[①]。

如图 15-2 所示，这个表单不仅能在错误发生的第一时间就做出响应，而且立即在该处使用了极为醒目的红色背景，并加以表意清晰的提示信息。

图　15-2

① 本书英文版发行后，该网站的注册表单已经被改进。网页为 http://eventful.com/join。——译者注

Eventful 网站并不只是把“User names must be at least 2 characters”（用户名至少需要 2 个字符）扔在那里就完事了。错误发生时会随即显示出更多的信息。具言之：

> **用户名必须：以字母开头；以字母或数字结尾；不能只包含数字；只能使用字母、数字、横线或下划线。**

已经说明了必须以字母开头，其实无需强调用户名不能只包含数字。尽管稍嫌啰嗦，但这些信息确实有助于解释用户在该处的可选范围，并且说明了应当如何改正错误。

我不喜欢 Eventful 表单的地方，同时也是绝大多数表单都存在的问题，在于它仅仅只在出错时才提醒你，而做对了的时候却什么也不说。哪怕正确地填写完整个表单，没有看到一个出错提示，你仍然会怀疑到底是不是真的完全没有问题。你仍然会惴惴不安地猜测是否在提交表单时将出现一个错误列表。

所以除了检测到错误时立即显示友好的提示信息之外，告诉用户“一切都没有问题”同样也很重要。

要把这些想法应用到 UpDownRepeat 网站的校验表单中去，我们从默认状态的表单开始，如图 15-3 所示。

图　15-3

我们希望只要用户正确地完成一个步骤，就在第一时间告诉他们。因此我们“剽窃”了 OurProperty.co.uk[①]的点子：用户正确地填写完某项后，在该项旁边显示一个绿色的勾号以表示“通过”。如图 15-4 所示。

---

① OurProperty 是一个英国房产交易网站。该网站注册页面为 https://passport.fubra.com/_ourproperty/signup.html。——译者注

Attendee first name: *

Robert

Attendee last name: *

Attendee email (e.g. me@mydomain.com): *

图 15-4

而下一步，毫无疑问，就是及时检测错误，并且友好地进行相关提示，以便用户改正。

为了做到这一点，我们在每个输入域的下方都创建了一个红色的文本信息，把输入域边框也设置为红色。这样能更加增强输入域和提示信息之间的联系。如图 15-5 所示。

Attendee first name: *

Robert

Attendee last name: *

Please enter your last name.

Attendee email (e.g. me@mydomain.com): *

图 15-5

而此处的关键在于我们应当在用户输入之后再显示错误提示。这是从 Remember the Milk 网站[①]学到的教训。在该网站中，输入域刚获得焦点就会出现错误提示[②]。

---

① Remember the Milk 是个人任务管理网站，能嵌入到 Google Calendar、Gmail、Twitter、BlackBerry 和 iPhone 等服务中。网址为 http://www.rememberthemilk.com/。——译者注

② 在该网站的注册过程中，每当用户试图在某个输入域中输入信息时就会出现一个红色的错误图标，并给出相应的输入限制。具体页面见 http://www.rememberthemilk.com/ signup/。——译者注

在用户还没有机会输入有效信息之前就责备他们把事情搞砸了，这是相当粗鲁的。我们希望避免这一点。因此只在输入域失去焦点时，我们才检测是否显示错误提示。

可能现在你的脑海中会跳出一个问题。

如果用户忽略某些必填项，直接跳到表单末尾试图完成填写过程，又会怎样？

在这种情况下，某些输入域根本不会获得焦点，因此提示信息也就不可能被触发。我们考虑到了这一点。

通常来说，人们会按顺序填写表单。所以我们有把握认为这个问题几乎不太可能发生。但为了保证在这种情况下用户仍然能纠正问题，我们建立了一个后备计划：在表单页面的顶部创建了一个奶油色背景的提示区域。

如果用户跳过了部分表单并尝试提交，他将会看到如图 15-6 所示的提示信息，同时在页面中列举了发生的错误。

Unfortunately, there was an error with your address. Please double-check this information.

The Art and Science of Web Forms

* = required field

Contact Information

You will receive a receipt along with the directions, time and additional details about the event once you complete your order. Please complete the required fields below.

图 15-6

假设用户看到了红色的错误提示并且修改了自己的输入结果，那么错误将会被绿色的勾号代替。如图 15-7 所示。

Attendee first name: *

Robert

Attendee last name: *

Hoekman, Jr.

Attendee email (e.g. me@mydomain.com): *

robert@miskeeto.com

图 15-7

这个表单也许还没有百分之百的做到错误核验，事实上谁也无法保证这一点，但是它为用户完成表单提供了坚实有效的反馈，这一点干得很不错。

实时更新信息

最后，我们希望在该操作中能清楚并正确地显示与会的费用。为了做到这一点，我们使用了 JavaScript 脚本，根据与会者和与会学生的人数及时更新总费用。如果有人享有折扣，就改变总金额。

如图 15-8 所示，在缺省情况下用户将会看到研讨会的标准费用（$149）。

Booking Information

Number of attendees:

1

How many of these attendees are students?

(Lowers price to $119. Make sure to bring each student's ID. They will be checked at the door.)

Promo code, if applicable:

Credit Information

Your total price of admission is: $149

图　15-8

如果用户把与会者人数改成 3 个，总金额会自动修改（$447）。如图 15-9 所示。

而如果与会者中有两名是学生的话，那么与会费用从 $149 变成了 $119，于是总金额会再度调整（$387）。见图 15-10。

Booking Information

Number of attendees:

3

How many of these attendees are students?

(Lowers price to $119. Make sure to bring each student's ID. They will be checked at the door.)

Promo code, if applicable:

Credit Information

Your total price of admission is: $447

图 15-9

Booking Information

Number of attendees:

3

How many of these attendees are students?

(Lowers price to $119. Make sure to bring each student's ID. They will be checked at the door.)

2

Promo code, if applicable:

Credit Information

Your total price of admission is: $387

图 15-10

在勾号图标、错误提示和即时更新信息的协作下，这个表单预防并侦测了可能在表单中出现的几乎任何错误。以上案例中，我们要达到两个目标。

第一是用户的目标——没有太多麻烦地完成表单。第二是重要的业务目标——让人们为研讨会付费。

在完成某项任务的过程中，那些用户目标和业务目标最好以愉快并且激发愿景的方式交互。用户所希望完成的，同时与业务考虑相吻合，这才是最理想的任务。上述的校验表单过程就是其中之一。用户最多只愿填写一个表单，而他们希望在这一次过程中就能完成购买。从业务角度出发，我们也希望用户达到同样的目的，因为这能给我们带来利润。

为了满足我们的需求，先得保证满足用户的需求。嗨——如果有人想给我们钱，那么至少我们应该帮助他来给，不是吗？

让我们面对它。填写校验表单并不像小镇一夜那么浪漫。尽我们所能来让这一过程更加流畅，让用户在输入账户信息的瞬间尽可能得愉快。

也许不能让这一过程很有意思，但至少可以让它别那么糟糕。

第 16 章 简化长表单

我曾与某家大型保险公司合作，为其人寿保险申请流程中所涉及的规章制度及合法性做了大量的研究工作。然后他们设计出了一套迄今以来我所见过的最具“欺骗性”的表单。

这个表单在一开始只有 3 道简单的是非题，例如“您、您的配偶及家属是否曾通过其他公司购买过 Acme 的专项医疗、牙科、眼科等保险项目？”

如果每个问题都选择“不”，那么你很安全。但如果其中有任何一道你选择了“是”，表单就会增长。新的表单域自动地嵌入到页面中，让表单变得更长，同时也需要你付出更多的努力才能完成。

比方说，如果你选择通过其他某个公司买了保险，就会出现关于医疗、牙科和眼科保险项目的复选框。如果你选中了医疗选项，表单将会再次增长，并且要求你填写该公司的名称、地址和电话号码。它还会询问你的保单号、生效日期和终止日期。如果你选择了牙科选项，则又会出现新的表单域。如果你 3 项都选了，那么你将会有一整个下午的快乐时间，试着回答一些可能自己都不知道答案的问题。

而这一切都是因为你在第一道是非题中选择了“是”。别忘了，另外还有两道题，它们都会出现类似的结果。也就是说，这个表单的总长度大概是我们现在所看到的 3 倍。

一开始只有 3 个简单问题的表单瞬间变成了一个醒着的噩梦。表单似乎无休无止：你的每一次点击都会让表单增长，就好像影片 *The Blob*[①]里面那个搞笑的果冻怪物一样。你可能会耗费一整个下午，而只是在尽力完成这一个页面。

---

① *The Blob*（变形怪体）是一部科幻 / 恐怖 / 喜剧片，Allied Artists 影业公司 1958 年出品。影片中一个如同巨大果冻的神秘外星怪物随着陨星落到地球上。——译者注

我的客户出发点是好的，他们希望用户能够在一个界面中填写完成，认为这样对用户会更简单。毕竟，等待载入一个新页面会非常让人讨厌，而且如果出现了错误，回退修正也会冒丢失会话的风险。

但是在用户努力完成这个表单的时候也失去了其他一些东西。换句话说，失去了还有可能完成它的所有希望。每次表单增长一些，你的心就会下沉一些。沮丧感很快就会出现，随之而来的则是焦虑。

而具有讽刺意味的是，这个表单完整打印出来其实也就只有两页纸那么长。只是因为网页的这种设计才会使它让人觉得无法完成。

我的客户希望知道如何能够改进它。我要了一杯双份摩卡咖啡、一大堆马克笔、一块巨大的白板，然后还有一根粗大的牛皮鞭。

好吧，我是有点夸张，但也只有一点而已。

16.1 清晰的预期

通过减少完成表单所需要的页面数（并且应用了一种我不太认可的渐进展现方式），我的客户成功地把所有东西归结为一个界面。但是与此同时，他们也使用户无法对整个过程建立任何一种期望。页面中没有任何东西能帮助用户预知将被这个罗网困住的时间。这并没有让表单显得更加简单，反而让人感到更加复杂。

我们的目标不应该是把所有东西都放在一个界面里。正确的目标应该是使用户相信表单并没有实际的那么复杂。不是欺骗或愚弄他们，而是创建一种简单的印象，使得表单给人感觉不那么复杂，从而在填写时不那么痛苦。

最后，我选用了一种设计，在整个过程的每一步都建立起清晰的预期使之更加友好，并且让整个表单感觉不像实际那么麻烦。

首先，我把表单分解成好几个页面，并且添加了一个步骤进度条，把整个表单变成了一个 Wizard 向导。这么做的目的是对每个步骤中用户需要填写的内容有一个清晰的指示，并且让用户随时了解还剩下多少个步骤。

其次，我去掉了多余的表单域，以避免用户在多处重复输入相同的信息。

然而，真正的工作其实是考虑如何在第一页就建立清晰的预期。毕竟，

如果得不到一开始几个问题的答案，我们也没办法告诉用户他们即将进入的是什么。

我以该表单原始版本中同样的 3 个问题开始。我的想法是第二页才是 Wizard 的第一步，如图 16-1 所示。

Do you, your spouse, or your dependents have any Group Medical, Dental, or Vision coverage through an Employer other than your coverage with Acme Insurance?

Yes

No

Do you, your spouse, or your dependents, have medical coverage with Medicare?

Yes

No

Do you, your spouse, or your dependents, have medical coverage with Medicaid?

Yes

No

Get started

图　16-1

很快我意识到这里的信息还不够以确定 Wizard 的样子。为了动态地创建 Wizard 向导，我不仅需要知道用户是否从其他公司购买过保险，还需要知道他是否购买了医疗、牙科或眼科的保险项目。因此我添加了一个复选框来获得这一信息，见图 16-2。

为了让事情更简单点，我决定如果用户选中了这 3 个复选框中的任何一个，那个 Yes 单选按钮也会自动选中。这样使得表单变成了一种“Conditional Form”。这种表单中，用户对某个选项的选择可能会使其他元素被激活或者失效。

Do you, your spouse, or your dependents have any Group Medical, Dental, or Vision coverage through an Employer other than your coverage with Acme Insurance?

Yes

Medical　Dental　Vision

No

Do you, your spouse, or your dependents, have medical coverage with Medicare?

Yes

No

Do you, your spouse, or your dependents, have medical coverage with Medicaid?

Yes

No

Get started

图　16-2

不过在做这个决定的过程中，我发现根本就没必要显示单选按钮。如果用户购买过这里面任何一种保险，那么答案自然是 Yes。如果他没有这些保险，答案则肯定是 No。因此我直接把第一题的单选按钮全部去掉，如图 16-3 所示。

延续这种思路，我意识到没有理由要用户去为任何问题选择 No。后续的 Wizard 向导过程只有在答案为 Yes 时才会增加内容。因此我决定摆脱掉所有剩下的单选按钮，并且把问题变成了复选框选项，如图 16-4 所示。

这样一来，那些没有从其他公司购买过保险的用户只需要单击 Get Started（开始）按钮，不需要其他任何动作。在原始的设计中，用户则必须要为每一道问题选择 No 的单选按钮。因此这个新设计需要花费的努力更少。

Do you, your spouse, or your dependents have any Group Medical, Dental, or Vision coverage through an Employer other than your coverage with Acme Insurance?

Medical　Dental　Vision

Do you, your spouse, or your dependents, have medical coverage with Medicare?

Yes

No

Do you, your spouse, or your dependents, have medical coverage with Medicaid?

Yes

No

Get started

图　16-3

Other insurance

I, my spouse, or my dependents has/have Group Medical, Dental, or Vision coverage through an Employer other than your coverage with Acme Insurance. *(Choose all that apply.)*

Medical　Dental　Vision

Medicare

I, my spouse, or my dependents has/have medical coverage with Medicare.

Medicaid

I, my spouse, or my dependents has/have medical coverage with Medicaid.

Get started

图　16-4

最后一个小花招，是第一时间就告诉用户在完成表单的过程中将会需要哪些信息。这样他们就能在开始填写之前就事先准备好，而不会在填写

过程中感觉受了欺骗。

为了做到这一点，我在表单顶部添加了一些说明性文字，把每一条必需的信息列成项目列表的形式①，这样更能鼓励用户快速阅读。见图 16-5。

If you have insurance through another provider, you may need to have the following information handy to complete this application:

- ***Statement of coverage*** *from your current insurance provider*
- ***Effective dates and/or end dates*** *for each type of coverage*
- ***Policy numbers*** *for each policy*

Other insurance

I, my spouse, or my dependents has/have Group Medical, Dental, or Vision coverage through an Employer other than your coverage with Acme Insurance. *(Choose all that apply.)*

Medical Dental Vision

Medicare

I, my spouse, or my dependents has/have medical coverage with Medicare.

Medicaid

I, my spouse, or my dependents has/have medical coverage with Medicaid.

Get started

图 16-5

乍看上去，新的设计似乎还没有最初的版本那样干净简单，但是现在完成它所需要的精力更少。同时一开始我们就为它建立了清晰的预期，而被划分为多个步骤也使之不再难以把握。

并不是所有的交互都能够简化。但我们却可以让它们看上去更容易。建立清晰的预期，并且把复杂的过程分解成能一口口干掉的小块，即使最复杂的交互也便于控制了。

① 该处的中文翻译如下。

“如果您从其他公司购买过保险，完成此表单可能手边需要有如下信息：

- 从其他担保方购买的保险项目详细信息
- 各保险项目的生效日期和终止日期
- 各保险项目的保单号”——译者注

第 17 章 让他们登录

我和 Kris Hadlock[①]以 33 Inc[②]的名义合作开发 Dashboard HQ[③]时，我们真的很希望把它做成一款伟大的、面向全世界的在线书签管理工具。现在我仍然相信这是一款伟大的工具。但当它发布时，网络社区正风行一时，而我们这个小程序并没有什么社区特性，因此它没有跟上潮流。最后我们还是取消了扩展 Dashboard HQ 的计划。

但不管怎么说，这款应用依然存在着。到今天我仍把它作为自己浏览器的主页面，而且它工作得仍然相当棒。

它的诸多好处之一就是登录流程非常简单方便。而在其他大多数 Web 应用中，通常让人头疼的是：你需要记住你的用户名和密码。

在 Web 应用中登录难就难在这里。如果碰巧你知道自己的用户名和密码，或者设置浏览器为你记住它们，那么这个过程非常简单。但如果发现其中有一个不正确，你可能就得费好大劲来找回你需要的信息，而这个过程可不令人愉快。

试图重新找回用户名或者密码时，很多网站都有一些多余的步骤。比如说，在使用了正确的用户名尝试登录了数次之后，最后你意识到自己真的忘掉了密码，于是单击了 Forgot Password？（忘记密码？）链接，然后被带到一个新页面，而在这里你必须再次输入你的用户名。

---

① 这位设计师在第 15 章已经出现过了。——译者注

② 33 Inc 是一个从事 Web 应用开发的二人团队，其成员也就是本书作者 Robert Hoekman, Jr. 和 Kris Hadlock。网站地址为 http://www.33inc.com/。——译者注

③ Dashboard HQ 是一款基于 Web 的软件，由 33 Inc 开发。用户能够把自己单位、家庭电脑和笔记本中的收藏网页都集中到一个单独的页面上，这样从任何终端都能通过网络进行访问。该产品也能让用户通过简单的操作对网页书签进行添加或编辑。网站地址为 http://www.dashboardhq.com/（目前国内无法登录）。——译者注

因为系统需要先知道你的用户名，否则它们什么都帮不了。而它们又无法确定你先前登录时输入的用户名是否正确。因此，你必须再次输入一遍。一旦它们知道了用户名，通常就会通过邮件把现有的密码发送给你，或者给你另外一个链接去创建新密码。

很麻烦，不是吗？

在试图改进这个古老的标准方案以便让它更有意思时，我们问了自己一个简单的问题。

如果系统已经知道了你的用户名，又会怎样？

17.1　再一次改进标准

如果系统已经知道了你的用户名，你便无需专门跑到另一个页面去输入它，然后告诉系统把密码发给你。因为系统能很快做到这一点，而不需要你付出任何额外的努力。也正因为如此，这个 Forgot Password ？的新页面也不是很必要了，并且登录页面只需一个链接就能把密码直接发送给你。

17.1.1　网络用户名与用户姓名

为了实现这一想法，我们对系统进行了设计使得用户名成为登录网站时使用的 URL 地址的一部分。

这样一来，用户收藏的不是首页，而是他自己的个人登录页。如果他的用户名是 John，那么他直接在 www.dashboardhq.com/john 的页面登录。如果在他登录后进入这个页面，那么将自动转向他自己的 Dashboard HQ 首页。如果没有，他只需要输入密码即可。看看图 17-1 吧。

如果碰巧某位用户和其他人共用一台电脑，他可能会发现自己正瞪着一个不属于他自己的登录页面。有一个小小的链接用于应付这种情况：I am not John（我不是 John）。单击这个链接时，本来显示的用户名将会转变成一个输入域，这样他就能输入自己的用户名了。

这里的想法非常简单。既然我们能做到这一点，为什么还得要求那些“熟客”记住自己的用户名呢？

不像大多数其他登录页面那样强迫用户每次都输入用户名，我们提供了一些额外的东西来减弱记住用户名的辛苦。



图　17-1

而正因为已经知道了用户名，因此某位用户单击 Forgot Password？的链接时，我们不再需要把他送到另一个界面再次输入自己的用户名。单击这个链接只会在后台给该用户记录中的邮箱地址发送一封包含密码的电子邮件，同时在登录页面显示一条简单的信息告诉用户密码已经发送。

17.1.2　更上一层楼

在 IxDA（Interaction Design Association，交互设计协会）[①]网站活跃的讨论组名单中，很多成员都把它当作自己的网上家园。设计这个网站的那帮家伙在简化登录流程方面迈出了更大的一步[②]。

实际上，这个网站甚至都不需要密码。

如图 17-2 所示，你所要做的只是输入你的名字和注册时所用的电子邮件地址。如果你记不住在注册时所使用的名字格式（只有名字、名和姓都有、或者其他什么组合方式）也不用烦恼。网站把你的电子邮件地址作为第一辨识规则，同时适配你在数据库存档中名字的任意格式。

① IxDA 是一个 2003 年成立的非盈利性专业协会，目前约有 1500 名来自世界各地的协会成员，致力于满足国际交互设计业界的需求。网站地址为 http://www.ixda.org。——译者注

② 该网站的登录页面见 http://www.ixda.org/profile.php。——译者注

图　17-2

比如说，我用 Robert Hoekman, Jr. 的名字注册，如果我在 Your Name（你的名字）一栏中只键入 Robert，我仍然能够登录。而如果我键入了一个我没使用过的名字，比如说 Joe，就会得到一条错误信息，如图 17-3 所示。

图　17-3

这让我感到在没有密码的情况下，自己的账户也还算有一点安全性。但如果我用讨论组其他成员的名字和电子邮件地址登录会怎样？我认识不少成员，所以这很容易办到。我能轻易登录并扮成那个人来发表文章吗？

呃，不行。当我试着这么做时，系统显示了一条信息表示我在之前从未用这台电脑或网段登录过，因此一条确认请求已经被发送到了我输入的那个电子邮件地址。

噢哦。希望我测试的那个电子邮件主人不要认为我在盗取他的账号。

IxDA 的解决方案并不一定适合所有的 Web 应用：它仍然不如密码那么安全。

同时，输入网络用户名是一个非常普遍的习性，很多登录的用户会在 Your Name 一栏中输入自己的网络用户名，而实际上他们需要键入的是自己真正的名字。违背了用户的通常习惯，将会使本已简化的登录流程反而比传统的设计需要付出更多的思考。尤其是用户并不知道可以用任意格式来输入自己的名字。这会使得他们犹豫是否需要把名字输入得完全与注册时一模一样。

但除了这些之外，IxDA 的登录过程是一次伟大的尝试。设计师们积极地挑战标准，不断探索如何改进，而不是犹豫不决。

标准绝不是永恒不变的。事实上，绝大多数时间里，在某些人刚树立起标准的下一秒钟它们就改变了。用崭新的眼光看待那些公认为陈腐的事物，这对于任何产品都是改进其用户体验的一种极佳方式。

第 18 章
计算字符数

打桌球时有各种方式能赢得比赛。最可靠的方式是日以继夜地磨练你的击球动作、纠正你的姿势、研读各种战术，同时坚持不懈地学习应对各种困境。这是那些职业选手的做法。他们花费数年时间反复训练，和那些更高水平的选手比赛，然后把自己推向更高峰。

而如果不那么稳扎稳打却同样有效的方式则是学习基础的击球、姿势和战术，然后成为一个有良好控制能力的不错玩家。在这里没有那种职业桌球选手的渴望，只是一种稳固的实力，然而它却能击败在酒吧挑战你的大多数对手。

最不可靠的方法则是绝大多数人学习并且一直尝试的，也就是 poke-and-hope（打了再说）方式。这意味着你猛地朝白球出杆，然后希望有什么球能碰巧落袋。

初学者基本上都依靠 poke-and-hope。更高级的桌球玩家偶尔在某种形势下也需要这种方式。职业选手最少，但也仍然发生。

和大多数桌球初学者一样，大多数电脑用户也没有学习过有效使用电脑的基础概念和技巧。这很自然，因为他们有很多比熟练掌握个人电脑更好的事情要去做，就像有很多比成为桌球大师更好的事情要去做一样。

不管怎样，这带来了各种各样的问题。人们在网站上右击，在弹出的环境菜单中寻找与任务有关的选项。他们在浏览器的地址栏里输入电子邮件地址，然后奇怪为什么他们的电子邮件不出现。他们在显示器屏幕前举着一张纸，然后奇怪为什么不能扫描。而更经常的是，他们在某个表单底部单击 Submit（提交）按钮，然后祈祷全部填写正确，这样就不会看到一堆讨厌的出错信息让他们感觉自己是个笨蛋。

换句话说，他们是在 poke-and-hope。

如果你是一个靠 poke-and-hope 瞎蒙的桌球玩家，也许就能够体会到大多数人使用在线软件的感觉了。在那里没有技巧、没有基础、没有深奥的知识；在那里只有殷切的希望，无论出杆怎样，都希望自己能更靠近一步，从而窃取 1~2 场胜利。

治疗这一切的良方就是 poka-yoke 机制（后文会继续详述）。

在 Web 应用中消除错误的可能性意味着 poke-and-hope 用户现在可以不再像低能儿那样使用产品了，即使他们几乎完全不知道网络到底是如何工作的，即使他们以为 IE 浏览器的“帮助”菜单也包含产品的信息，即使他们在网站的搜索栏里输入 URL 地址然后期待新网站会自动出现。

改进 Web 应用所能做的最重要的一件事就是减少出错的可能性。现在如此，以后也一样。不会出错的用户感觉自己聪慧、备受尊重、有成效。

聪慧、备受尊重、有成效的用户是整个世界上最好的用户，因为他们会告诉其他人使用你们产品时的美妙感觉。这是最好的市场营销。

如果你希望人们也能这样谈论你的产品，那么一开始最好的办法就是把 poke-and-hope 转变成 poka-yoke。

我尝试在每一款产品的设计中都做到这一点，因此当我坐下来写这一章时，手里握着一大堆例子。我选择了一个很常见的问题做例子：带字数限制的输入域。

18.1 到达限制

有很多表单域都需要字数限制。比如说，在某个创建密码的输入域中就只能输入 8~10 个字符。

每当我需要创建一个带字数限制的表单域时，几乎都会使用同样的方式。因为在我第一次设计表单域时，便游历了各种不同的网站，研究我所能找到的每一个类似情况，并且记录它们的工作方式，最后得出我认为是最好的设计方案。而当自己的方案并不合适时，我仍然需要去寻找其他一些实例。

我知道你正在疑惑为什么我会纠缠于这样一个简单又平凡的细节，对吧？毕竟这不是一个很困难的界面问题，甚至它都不值得真正去考虑。

但在你下结论之前，想想以前曾遇到过的那些受限制的表单域吧。

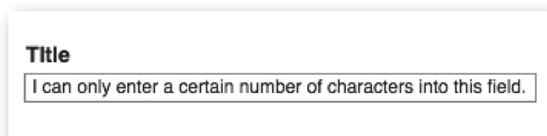


图　18-1

如图 18-1 所示，你在某个受限表单域中输入了一条完整的字符串，提交表单，然后看到如图 18-2 这样一条信息：You can enter no more than 50 characters.（你只能输入 50 个字符。）

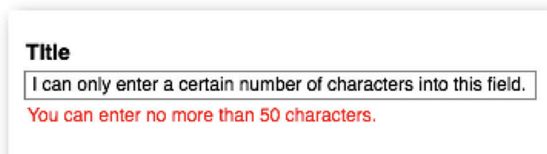


图　18-2

自然，这个限制会迫使你一个字一个字地去数已经输入的字符，然后考虑如何调整以减少字数来满足条件。而这是一个冗长、乏味又糟糕的事情，还不能不做。于是你感到愤怒，而且由于不知道空格是否应该计算在内，你对自己的计算结果是否精确也毫无自信了。

而另一种常见的情况是，你快乐地一直打着字，结果却发现键盘出了毛病。

如图 18-3 所示，你键入了整整一篇小说，结果却发现最后 200 个字符根本没有输进去。为了找出是哪里出了毛病，也许你会打开另一个程序输入某些东西以确保你的键盘仍然正常，也许你会到电脑背后检查键盘的 USB 插口是否松掉了，我自己就曾经干过。

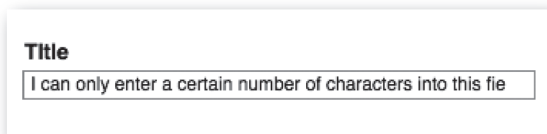


图　18-3

就像这样，一个普通的输入域会让你怀疑整台电脑发生了故障。

悲哀。

这就是你想为用户创造的瞬间吗？我不这么想。幸运的是，这个问题很好解决。

18.1.1 编辑中的 poka-yoke

我们只需要一个简单的方式让用户知道发生了什么。首先，让他们知道这里有字数限制；其次，让他们知道何时会到达这个限制。

poka-yoke 可以拯救你！就如我在 *Designing the Obvious* 一书中所说，poka-yoke 是一个日语词汇，意思是“防止错误”。其含义有二：侦测机制和预防机制。我很乐于找到办法在网上应用这个想法。

侦测机制就是让用户知道有错误发生，并且使他能够立刻修正。各种出错信息都属于 poka-yoke 的侦测机制。

而预防机制则表示在一开始就避免错误发生，并且受限表单域就是实践这一想法的好地方。

最简单的解决方案就是加上一条说明性文字，向用户解释这里的字数不能超过 50 个字符。如图 18-4 所示。

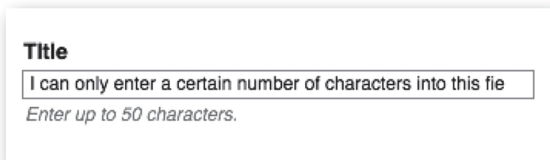


图 18-4

这能让用户知道限制条件，所以他们就不会疑惑为什么键盘突然失灵了。但如果在到达限制时没有任何提示，他们仍然需要计算字符数：

“‘I can only enter’……让我们看看……这里有 13 个字符，所以……我还能再输入 37 个。好……‘a certain number’……，现在多少个了？还有空格应该算进去吗？噢，真讨厌。”

为了在这段说明文字中加强 poka-yoke 机制，我把文本中的数字 50 改成了一个不断减少的变量。

如图 18-5 所示，现在当用户输入时，每增加一个字符这个数字就会减少。用户能在整个输入过程中确切地了解自己还能输入多少字符。当然，如果删除字符，这个数字就会增加。

图　18-5

见图 18-6，即使他到达 0 后，也仍然能继续调整输入的文字。

图　18-6

整理好这一设计之后，我立刻将其设制为一条标准，还发表了一篇关于它的博客。

而就在那时有人指出这个设计还有点小问题。

对于使用屏幕阅读器的用户来说，因为屏幕阅读器是按上下顺序来阅读页面的，所以这些用户只可能在出现问题之后才会发现这个关于字数限制的信息。

我真笨。

要弥补这一点，我仅仅只需把这个不断减少的变量挪到表单域的上方，见图 18-7。

图　18-7

没错，使用屏幕阅读器可能需要来回地使用 Shift → Tab，不断地把焦点切回到该信息上以得知还剩下多少个字符，但至少在输入之前就能了解字数限制条件，而且能随时检查当前的字符数。

18.1.2　一点小警告

顺带一提，Twitter 网站[①]在设计字符计数方面走得更远。一开始不断减少的变量是灰色文字，而一旦用户输入字符数超过了 120，该变量就会变红，提醒他们还剩下不到 20 个字符了。

干得很棒。

即使用户在一开始不会注意到这个文本，但是通过颜色变化这种方式也会在输入时引起他们的注意。红色能在绝大部分网页中凸显出来，而且突然出现的红色通常能打断用户的输入，让他们注意到字符计数器。

我对该设计唯一的问题就是，这个字符计数器本身没有相关的文本或标签来解释它。如图 18-8 所示，只是一个单独的数字放在输入域的右上方。

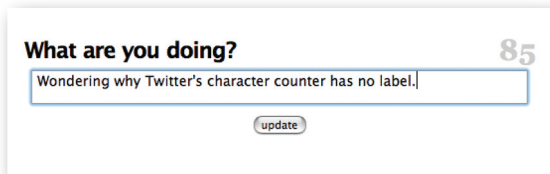


图　18-8

不管怎样，加入简单的字符计数器便能把一个潜在的错误转变成另一个“原来使用这个产品只需常识”的瞬间。

不再迫使用户自己逐字计算字符，也避免了他们在看到出错信息前白白输入过多，只需添加简单的字符计数器，这个交互就一目了然了。

① Twitter.com 是一个能让朋友、家庭和工作伙伴之间通过记录日常琐事而随时保持联系的在线服务。网站地址为 http://twitter.com/。——译者注

第五部分

参　　与

互联网，其中很大的一部分乐趣在于参与。

我们以前只是被动的浏览者，现在却可以和其他人联系交流、管理和编辑自己和其他人的信息、共享想法、制定计划，以及其他各种好玩的事。

而今天你有无数种方式可以加入进来，并且网络社交技术的提高也使得我们可以做任何事情。你可以和身处日本的小孩打扑克，也可以和正在 1000 英里外出差的朋友随时保持联系。

不过，尽管用户可以尽兴地为博客写回复、在 Facebook[①] 上交朋友，或者通过某个网站构建工具为自己设计网页，这种种瞬间中都充斥着混淆、丧失自信的可能性。

在这一部分，我会讲述一些关于参与的故事，并讨论一些原则，希望能帮助你们为用户把这些瞬间变得更积极和易于理解。

后面 6 章中，我会论述为什么说“清楚”比“有效”更为重要、如何鼓励和激发社交行为，甚至说为什么“跟从”人们在网站的动向会比成为他们的“朋友”更加合适。

不管潜在的目的是什么，哪怕是最复杂的交互，也能变成令人愉快的瞬间。

① 众所周知，Facebook 是一个通过上传照片来结交朋友或与亲友保持联系的在线服务，目前其范围扩展到了视频、游戏等方面。网站地址为 http://www.facebook.com/。——译者注

第 19 章 创建个人资料

在为第 1 章中平面设计师社区设计时，我遇到的一个挑战就是要构建一套更有意义的用户个人资料系统，而不仅仅只是包含网站成员信息的简单页面。

我所做的并不算什么革命或创新，但它是一个很好的例子，说明了应用程序本身的设计确实能够影响到用户所需的信息类型。

在很多网站里，用户的个人资料包含了他们的一些重要信息，比如他的名字、网站 URL 地址、即时消息信息，可能还包括他的兴趣爱好，以及为什么加入这个网站的简介。这种设计对于其他网站成员来说并不一定很有帮助，因为它只是存放了该用户的这些信息而已，成员相互之间并没有得到推介。然而一定程度上的温暖感仍然能诱使一些新用户花费宝贵的时间输入这些细节信息。

然而，在这个平面设计师的社区中，我们知道有一些不同种类的用户，比如设计师，以及那些需要设计师合作的人（客户）。我们还知道，经常有些用户喜欢研究其他用户收藏的最喜爱项目案例、评论回复以及个人上传的案例，希望这些能让他们看到更多好的内容。比如说，当 Josh 很中意某个项目案例时，他很可能还希望看到该设计师的其他案例。同样地，Josh 可能还会注意到另一位客户 Sarah 在某个项目上给的费用很高，那么他也许会想看看与 Sarah 合作的那些设计师。

为了满足这些需求，首先我们将创建资料的过程**分层**，然后设计了一种能自动变化的个人资料页面，让它能动态地记录用户在网站上的行为，就好像汽车的仪表盘一般。

让我来解释一下。

19.1　递进地增加

决定在本网站注册的每一位用户都会被要求填写基本的个人资料信息，它们包括用户的姓名、电子邮件地址、密码以及信用卡信息。如图 19-1 所示。

图　19-1

然而，平面设计师则必须再做一些其他用户不需要做的事情，例如创建个人案例集、标明工作费用以及职能范围，还有指定自己的专业领域。

为了做到这一点，我们使用了**分层的**方式。任何创建案例集的人都需要填写一些额外信息，而填写的内容将会添加到他们的个人资料中。这些信息只面向那些试图通过本网站推销自己的设计师。如图 19-2 所示。

因为只有平面设计师才需要输入这些信息，所以我们把它们放到了注册程序之外。简单来说，我们不希望登录用户感觉只有设计师才能注册。

图　19-2

那些招募设计师进行项目合作、或者只是浏览网站找寻灵感的人，将不会看到要求填写这些附加资料的表单。

19.1.1　把数据变成仪表盘

不管用户属于什么类型，设计师、客户、或者爱好者，系统都会自动为每位用户创建一个个人资料页面，并且其他所有网站成员都能看到。

当用户在网站里有所举动时，个人资料页面就会相应地更新以便反映他的行为。

如图 19-3 所示，如果 John 创建了自己的案例集、收藏了其他人的案例集、标明了职能范围和工作费用、发表了博客文章等等，他的个人资料页面就会相应更新以便显示这些信息。

任何网站成员都可以浏览其他成员的个人资料，了解他喜欢哪些案例、创建了什么案例或者评论了什么案例，还能阅读他的博客文章。

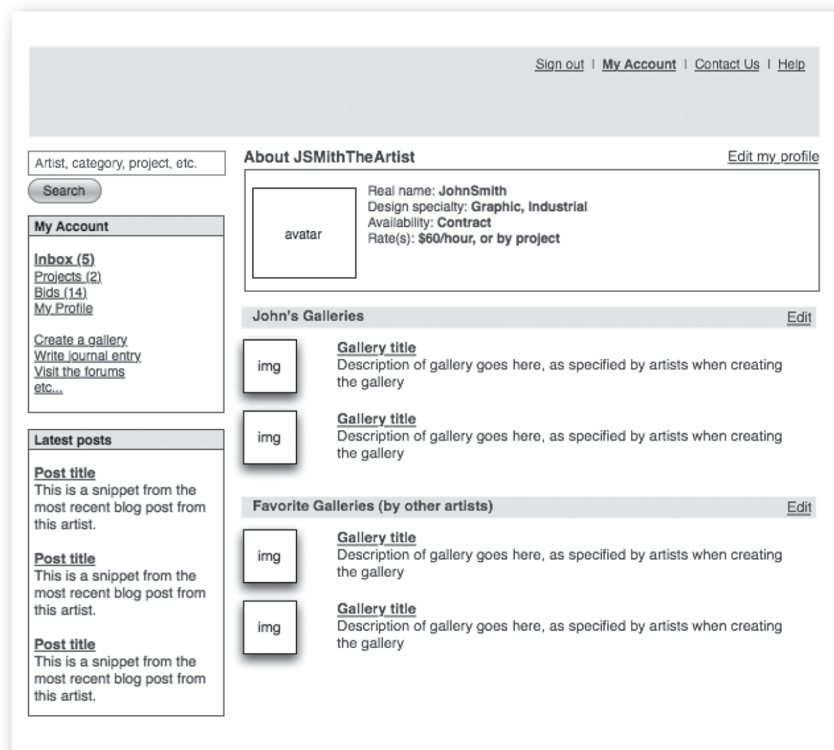


图 19-3

这种个人资料页面展示的内容和固定资料页面相比，明显要更加充实和生动。它变成了一个栩栩如生的用户行为记录。

这样做有很多好处。首先，只要找到了中意的案例，就能马上点击查看该作者的其他案例。其次，当看到有符合自己品味的评论或标价时，你也能很方便地查看该作者喜欢的其他案例以及他对它们的评价。

换句话说，你可以把任何用户的个人资料页面作为寻找优秀内容的入口。如果你发现另一位客户 Sarah 喜欢的东西和自己非常一致，你便可以通过 Sarah 的个人资料看看她最近查看过的东西。

19.1.2 空白资料页面

如果用户注册后没有任何行动，那么他的个人资料就会是一个**空白资料页面**。

空白资料页面里面没有任何数据信息，只有该用户出现某种行为才会添加。由于目前这位用户在网站里还没有做任何事，例如收藏某位设计师的案例，所以他的个人资料里面几乎空空如也。

空白资料页面的好处在于它们可以鼓励人们采取行动。我们这个例子中，没有个人资料信息的资料页面会提供一些链接，方便用户浏览各个案例、创建案例集等等。这种方式将更能刺激尚未采取行动的用户开始创建自己的案例集，或者发表评论。

个人资料通常包含一些用户简单的个人细节信息，但个人资料也能提供某种机会。在这个例子里，个人资料变成了一种有效的人际交往手段，提供了一条清晰的途径来了解其他成员的喜恶。

通过提供这种信息，用户将开始信任与之品味类似的成员，并且会时常回访，获知他们最近的浏览内容。也正因为这种信任，他们也有理由一次又一次地反复光顾这个网站。

第 20 章 编　　辑

在第 3 章，我谈到曾经重新设计了一个在线网站产品的导航。本章我想谈谈同一个产品的另一个问题——编辑功能，这可比稍稍修改一下导航菜单要麻烦得多。

这个产品就是多功能症的负面典型。在科学术语中，多功能症是指产品由于包含过多功能而引起的一种可用性低下的疾病（好吧，这并不是很科学）。

在我们这个例子中，一大堆的功能被塞到产品中，造成了极为繁复的用户体验。而这一点在它的编辑功能（该产品最基本、最常用的功能）中则尤为突出。

只要用户试图执行某个编辑任务，不管是为某段文本调整格式还是在页面中插入图片，系统总会弹出同一个窗口。而几乎所有的交互都得和这个讨厌的弹出窗口打交道。

这个弹出窗口包含了文本格式工具、图片编辑工具、插入时间戳、创建 RSS 的 Feed[①]……只要是你能叫出名字的，都在里面。

很自然，我非常疑惑为什么不能把这些工具放在主界面里，从而去掉弹出窗口。而同时我也很疑惑何苦让用户每次都得在这么一大堆工具中费劲地找寻自己需要的那一个。

---

① RSS（Really Simple Syndication）是一种用于共享网上新闻频道、博客和其他 Web 内容的数据交换规范。这种描述和同步网站内容的格式是目前使用最广泛的 XML 应用。用户订阅了某个站点的 RSS 之后，不用登录该网站就能通过自己的浏览器或 RSS 阅读器随时了解该网站的更新情况。也有人说 RSS 的全称为 Rich Site Summary（丰富站点摘要）。所谓 Feed 就是指 RSS 的源，其中包含的信息能直接被其他站点调用。——译者注

比如说，要想为文本设置格式，用户必须把鼠标移动到文本上，在随之出现的小菜单中选择 Edit（编辑），然后等待弹出窗口出现并载入各种工具。这些完成之后，他必须先在弹出窗口中选择想编辑的文本，然后在工具栏的众多图标里找到想要的那个，例如 Bold（设置粗体）。确认了格式变化之后，他还得先单击 Save（保存）按钮，等待弹出窗口关闭，然后再等待主界面里面的文本更新。

如果这位用户想修改另一段文本的格式，他必须重来一次。如果想撤销这次改动，他唯一的选择也是重复同样一系列步骤。

正如你所想象的，在进行几次修改后这个过程就会变得相当单调乏味。无论如何这些都称不上令人愉快的瞬间。

而随后的两轮可用性测试也证实了这一猜想。我们的测试者连最基础的任务都失败了，其中一些测试者还不是第一次使用这个产品。

绝大部分情况下，这个弹出窗口里所有的编辑工具都能够使用。导航用户在其中很难找到完成手头任务所需要的工具。整个弹出窗口里面大概有一百多个不同的工具和选项，但其中只有 3~4 个能真正满足所有的任务。比如说，修改文本格式时，插入日历图标就没有用，甚至还会妨碍用户。而这个弹出窗口中的大部分工具都和最常用的任务无关。

而把所有这些工具放在一个弹出窗口中，这意味着迫使用户脱离了任务环境本身。他们无法看着页面进行编辑，因为弹出窗口挡在了中间。

由于资源缺乏，在我加入之前，这一项目没有交互设计师参与过开发。程序人员在一开始尽了自己最大的努力（尽管用户和内部管理层不停地要求添加功能），时刻谨记以保证用户的兴趣为出发点做出决定。然而，缺乏对可用性的深刻理解和产品发展的长远计划，最后导致开发者不得不被迫往里面添加一个又一个的新功能。同时因为缺乏整理头绪的时间和有效的执行力，开发者落入了陷阱。他们唯一能做的就是在界面里找寻空余的地方塞入新功能，然后祈祷能凑合过去。

关于这个产品他们听到的抱怨和我一样，也和我一样对此感到厌倦。幸运的是，我们在几个关键之处达成了共识，从而能够开始进行再设计。

我们从编辑功能着手。

清除这一片混乱并不容易。项目开发者以及我团队里的其他设计师花费了几天时间进行讨论和白板会议，最后提出了一个技术上可行、又能解决大部分显著的可用性问题的方案。

20.1　在正确的时间显示正确的工具

要想减轻编辑功能的混乱程度，最好只显示与当前任务相关的工具，同时我们也希望用户能直接地编辑他们的网页，而不是被带到另一个界面。

在这个过程中，我们希望能让这个功能超载的产品看上去更简单一点。尽管它几乎无所不能，但我们相信，在正确的时间显示正确的工具，能让每个任务更加容易。

我们把所有的工具和选项从弹出窗口上挪出来，放到了主界面中，就在被编辑网页旁边。这个举措解决了任务的环境问题，但最棘手的难题还在后头。

怎样才能隐藏那些和当前任务无关的工具和选项呢？

我们曾考虑简单地把工具置灰，让它们无法使用。但是这意味着用户将不能在中途改变主意。比方说，如果在用户选中好文本时把图片编辑工具置灰，那么本来想编辑文本的用户就无法在中途转而去插入一张图片了。

因此，在隐藏无关工具的同时，我们还必须确保所有的工具在任何时间都能被用户找到。

我们考虑了一种设计，类似于在很多 Adobe 产品中用到的 Properties（属性）面板。例如 Adobe Flash。见图 20-1。

这种面板上的选项会根据用户当前使用的工具而改变 [图 20-1 中显示的是 Flash 里 Text Tool（文本工具）选项]。

它可以办到只显示与任务相关的控件，但我们不能让屏幕上满是这些控件面板。所以这种设计不能保证用户能随时使用所有其他的工具。

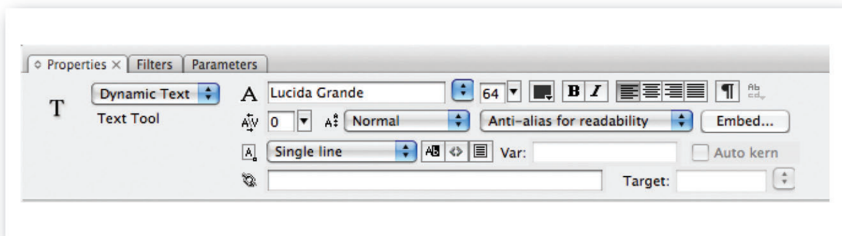


图　20-1

幸运的是，我们知道可折叠面板控件（在 14 章讨论过）。在进行这个项目的同时我们还有其他几个项目，其中某几个就用到过这种可折叠面板，所以我们想把它也作为一个选择考虑进来（为了保持图片更容易辨识，我只在这里展示了可折叠面板和编辑区域。要想展示整个界面恐怕得需要更大的一本书）。见图 20-2。

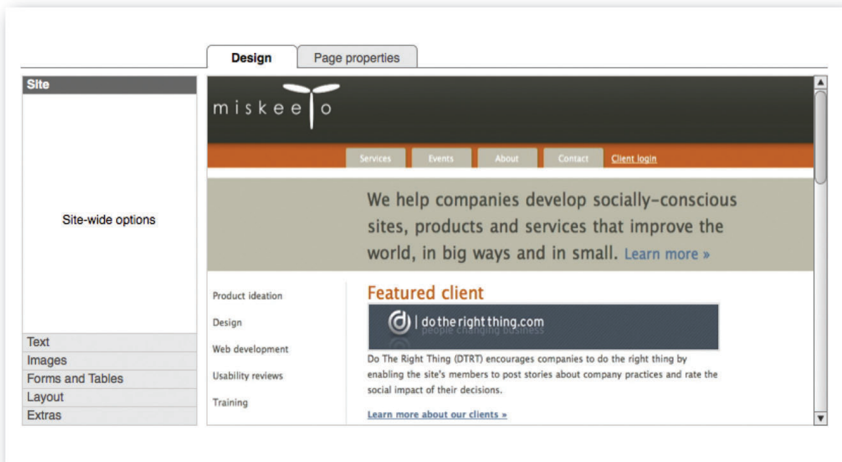


图　20-2

从这个粗略的草图中可以看到，我们能在可折叠面板中将最常见的工具和选项表面化。而根据用户不同的行为，相对应的面板便会自动打开。

例如，用户点击某个文本段时，Text（文本）面板将会自动打开。见图 20-3。

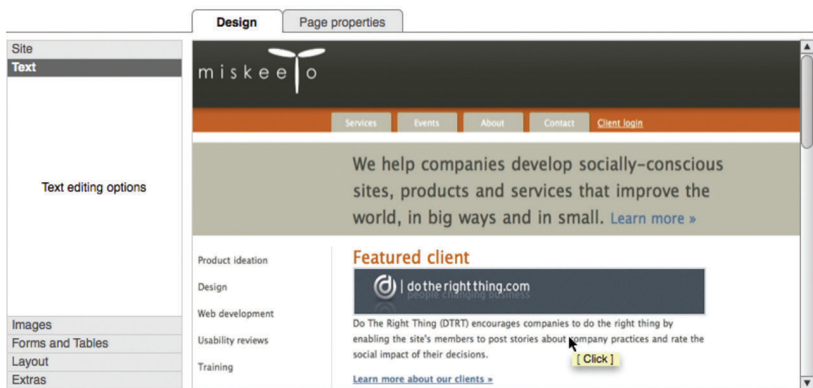


图 20-3

这个面板将会提供格式选项，例如 Bold（粗体）、Italic（斜体）、Underline（下划线）等，它同时还包括一个菜单，可以用于选择字体、字号以及字体颜色。

这些就是在正确时间里显示的正确工具。用户点击某个文本段时，不管他们是想写、修改还是想设置文字格式，这些都是能够方便取用的工具。

用户点击图片时将会打开 Images（图像）面板，展现编辑图片所需的选项。点击网页中的空白区域则会打开 Site（站点）面板，出现的选项可以应用于整个站点，例如 Create Page（创建新页面）和 Organize Navigation（组织导航栏）。

我们花了数天时间弄清楚在每个面板中应该放些什么工具，以及如何为用户提供不在面板中的工具。

解决办法也很简单：应用递进显示（Progressive disclosure）原则。

20.1.1 把高级的玩意藏起来

正如在第 11 章和第 16 章中讨论的，**递进显示**是一种根据功能是否普及或必要逐步展现它们的方法。

这个例子中，只显示与任务相关的工具一点错也没有。但我们也仍然需要一种方式用来提供其他工具或选项，以备不时之需。

所以我们设计了一种文档模式[①]对话框，用它来容纳附加的选项。**文档模式**是说，对话框将短暂地阻止用户和被编辑网页之间的交互，即无法使用 Edit（编辑）面板，但用户仍然能够与应用程序的其他部分以及浏览器本身交互。

比如说，用户点击某个文本段后，打开的 Text 面板里包含了一个将选中文本变成超链接的选项。但我们不能把与超链接有关的所有选项都放在可折叠面板里面，而是把它们挪到了一个对话框里。

所以，Edit 面板变暗而且不可用，同时这个对话框在同一个屏幕区域打开。在对话框里，选择超链接的相关选项，然后单击 Finished（结束）按钮创建这个链接。见图 20-4。

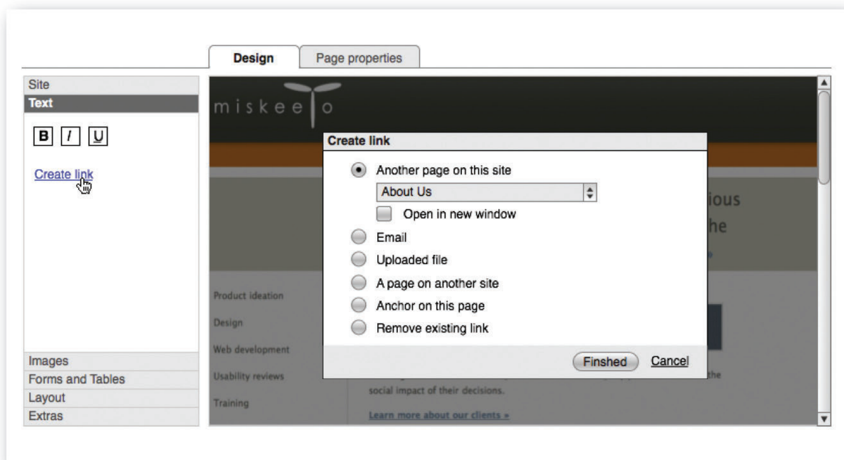


图 20-4

交互的触发器 Create a link（创建链接）选项出现在可折叠面板中，而该功能的选项则出现在对话框里。

① 与应用程序模式对话框不同，文档模式对话框将阻止用户和当前文档窗口发生其他任何交互，但用户可以切换到该应用程序中的其他文档窗口，或者切换到其他应用程序。读者可以通过以下链接查阅 Apple 的人机界面指南（Apple Human Interface Guidelines）中关于 document-modal 的详细说明 http://developer.apple.com/ documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGWindows/chapter_18_section_7.html。——译者注

20.1.2　打扫干净

在写这一章时，还有不少人在使用这款产品。从功能上来讲，它还是非常缓慢、笨重。毕竟创建网站可不是件轻松的事，但我们对于编辑功能的检查在相当程度上仍然改进了产品的可用性。

在某个应用产品中，如果你发现要完成某些任务时会牵扯到非常多的工具和选项，那么最好是尽快以全新的眼光重新检视，然后设法加以改进。在我们这个例子里，递进显示就是关键：只显示每项任务需要的工具和选项的关键。

关键就是在正确的时间显示正确的工具。通过这种方式，完成任务所需要的步骤和用户需要做出的决定都大大减少了，这让每一次交互都更容易执行了，而且每一个交互的瞬间都令人更加愉快。

第 21 章 开展社交活动

啊，“朋友”，一个多么美好的词汇。我们和朋友一起笑，一起哭，一起偷偷开小差。远离电脑的时候，正是这些人陪伴着我们共度日益宝贵的分分秒秒。

但是通过网络，例如 Facebook、MySpace、LinkedIn、Twitter 和其他许多全民媒体网站，我们所建立的朋友关系与日常生活中建立的朋友关系又稍有不同。

网络的朋友关系通常带些做作和人为。当然，我们也可以像在同一个网站里那样和真正的朋友去交往，但更常见的是，我们和那些不认识的人结交。他们可能是我们在论坛里、讨论组中认识的，也可能是朋友的朋友，或者是我们素未谋面、但一直尊敬和倾慕的人等等。

而有些悲哀的是，这种朋友关系通常没有什么意义或深度。拿 Facebook 举个例子。当你在 Facebook 上和某些人结交为朋友时，他们的行为都会列举在你的 Facebook 页面上。但是他们的行为往往又都是和其他人交朋友、赠送虚拟礼物、发消息等等。当然这很有意思，但是最后你得到的并不是真正的信息，而只是一堆噪音。这可能乐趣无穷，但同时也毫无目的，至少事实上如此。

就这一点而论，交友和其他行为一样，环境不同，情况也不同。

对于 Facebook、MySpace 等网站来说，交友是他们业务开展的基石，但是许多环境下的交友其实毫无意义。

当 DoTheRightThing 那帮家伙决定尝试在站点里加入交友系统时，我猜想这不过是又多了一个不太成功的例子而已。

DTRT 创立的目的是想提高人们对一部分企业道德或不道德举措的意识。这并不是一个很具娱乐性的活动。当然，这种社会意识的行为可以令人感觉好玩或者满足，但肯定不至于像一场周末狂舞夜那样。

他们的基本想法和其他许多社交网站一样，就是用户能够通过其他成员的个人资料页面交友。访问该页面的用户单击 Become My Friend（成为我的朋友）按钮，然后两人之间立刻就会产生一段持久的、意味深长的友情。他们可能会在周末一块参加街区扑克比赛，他们的儿子可能会成为最好的朋友，并且上同一所学校，最后在集体婚礼上迎娶一对姐妹。

唔，所以我不是很理解 DTRT 里面交友系统的目的。

我的第一个问题是，当你成为某个人的朋友时会发生什么？

结果是这个想法太新了，新得还没有人能真正充实它。他们知道自己想鼓励人们在网站里浏览更多的故事，然而事实上成员通常只会看一个故事，评分然后离开。同时他们还认为提供一种方式供人们沟通是种很好的刺激。

我觉得他们在这里已经有了初步的概念，但仍需要一些调整。

21.1　朋友与随员

Facebook 提供了无数种方式让人们随时掌握朋友的动向，MySpace 也是如此。然而虽然大部分这些功能和动向都很不错，但在 DTRT 这种关注社会变迁而不是娱乐的网站上却一点用没有。

在 DTRT 上添加交友功能后正如图 21-1 所示这个样子。

而我们也正是从这里开始。页面中有一个单击与该用户结交朋友的按钮、该用户部分朋友的头像，以及可以查看他所有朋友的链接。

但是这个页面与 DTRT 的感觉并不是非常贴切。因此我和 Rod Ebrahimi（DTRT 的创始人）[①]花了好几天通过邮件相互发送线框图，并且一直讨论在线交友的本质，以及这个想法如何才能适配 DTRT。

---

① Rod 在 DTRT 网站的个人资料页面为 http://dotherightthing.com/users/rod。——译者注

图　21-1

我们几乎得不出答案。

某人单击了 Become My Friend 按钮后会发生什么？什么样的行为才会增进这种朋友关系？这些东西是否真的能鼓励网站成员去阅读或发表更多的故事？

不管我们怎样看，交友功能对 DTRT 来说还是没什么意义。

我们真正想要的并不是建立友谊，而是一种简单而又不张扬的功能。它能激发人们阅读更多的文章、与网站发生更多的联系。

最后，我们跟从了 Twitter 网站的领导。

Twitter 提供了一种较为弱势、被称为 Follow（跟从）的交友方案。简单来说，任何网站成员都可以到另一个成员的个人资料页面上单击 Follow 订阅他在 Twitter 的 Feed。

不用付出太多精力，而同样能随时掌握另一位成员的动向。这样就够了。

我们认为“跟从”某人动向这种简单的行为更适合 DTRT。通过查看自己跟从的人发表的文章，或者寻找那些有很多人跟从的人，你可以很快找到感兴趣的故事、发现谁给哪个故事评了高分等等。

有意思的是，要想扭转形势只需在左边的工具栏里面改几个字就行（因为真正的交友系统尚未构建，因为没人知道该构建些什么。通过线框图，我们可以在写代码之前做出任何决定，也可以随时修改）。

如图 21-2 所示，我们把按钮上的文字 Become My Friend 改为 Follow My Activities（跟从我的动向），然后修改周围的文本以配合这一改变。在其他成员头像那里，我们用 Who’s Following Me?（谁在跟从我？）代替了 My Friends（我的朋友们），用 See all 46 followers（查看所有 46 位跟从者）代替了 See all 46 friends（查看所有 46 位朋友）。

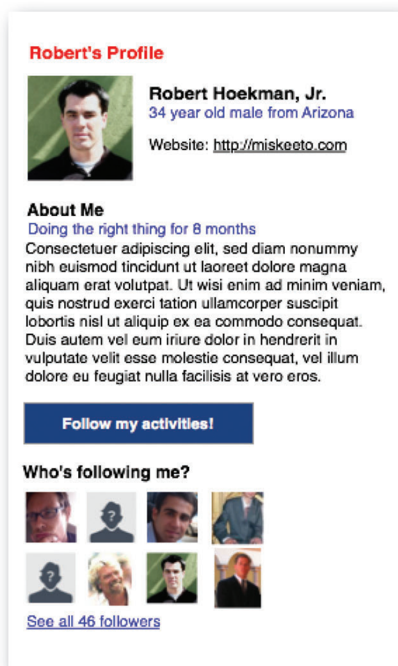


图　21-2

这些简单的改变让事情变得顺眼多了。现在，DTRT 成员在个人资料页面简单扫一眼就能看到其他各种人在网站上的行为。理论上来说，这能够激发他们去阅读或发表更多的故事。

不在场时的照片

在写这一章时，这个系统还没有被构建好而尚未添加到网站中，所以我不能断定在那一刻我们是否做出了决定。但我们已经仔细研究了问题，并且根据领悟力做出了决定，因此我们相信网站肯定会变得更好，而且我们将能为 DTRT 的用户提供一些真正适合他们的东西。

我知道全民媒体网站在当今如火如荼，然而随之而来的那些功能，如交友、评分、投票、评论等等都需要仔细周全地考虑，而不要随便就把它们像大杂烩一样扔进社区网站。这对于 Ajax、DHTML、Adobe Flash 以及其他热潮都是一样。

不要随便就被热潮牵着走。请三思而后行。

第 22 章
显而易见的博客

即将展现给你们的这个博客模板，并不是为某个具体客户创建的。实际上，我没有使用设计软件，没有用白板或黑板，也没用到笔和纸。甚至在开始设计时，我连具体的想法都没有。

故事是这样的。

2007 年 7 月，我在 WordCamp[①]做了一次演讲。那次是由 WordPress 的开发商 Automattic 主办的一次年会，很多 WordPress 的博主聚集一堂，在两天的时间里以博客为主题进行探讨。

抵达时，我已经准备好了演讲的内容，主要基于我写的 *Designing the Obvious* 那本书。之前我参加过好几次类似的演讲，对自己要说的内容相当自信，而且已经修改了幻灯片的内容，把重点更多地放在了博客上。所以我认为这个一定能满足听众。但是在会场看了几个演讲之后，我意识到这些听众可能不会太关心我准备的那些 Web 应用设计的指导原则。

我即将在三百多人面前站起来，然后被炸成碎片。而现在还有大约 1 个小时来避免这个迫近的危机。

我溜出大厅跑到了休息室，打开了笔记本，快速地细读了一遍自己的幻灯片。

我做了一点改动，然后绝望地试图劝服自己一定能侥幸成功。

---

① WordCamp 是一种非正式的会议，主题聚焦于博客在线软件 WordPress（见本书第 1 章）。这种会议由该软件的用户（或者说博主）自主发起，可以和开发人员共同针对该软件的各个方面进行探讨，并且把会议概况发布到 WordCamp 中。WordCamp 的网址为 http://central.wordcamp.org/。——译者注

我在骗谁呢？

我关闭了幻灯片，打开了文本编辑程序，开始记下一些笔记，内容是如何才能做好一个博客设计。

大约 30 分钟后，我走上讲台，告诉不停聒噪的听众我扔掉了自己的讲稿，也不会给他们展示什么幻灯片。听众鼓起掌来，很明显没人喜欢幻灯片。

在后面的 1 个小时，我引领了一场关于什么才是优秀博客设计的讨论，而不是演讲。我论述了一些自己书里面的原则，然后提出了一些开放式问题，范围集中在如何把这些原则应用到博客设计里面去，以及怎样在这些想法的基础上再更进一步。

幸运的是，听众跟上了我的步调。他们声音很大，而且投入、积极、情绪高涨。他们喊出了不少好主意。他们在整个大厅里争论。他们简直令人惊异！

讨论时，我用脑子记下了大家共同认可的每一条。演讲的最后，我把所有这些意见总结出来，最后一次引用了我的书，终于安然无恙地走下了讲台。

几个小时后，我打开了 OmniGraffle 创建了一个基于刚才讨论的线框图。

22.1 通往更好的博客的三条路

那次讨论中主要得出了 3 个主要结论。

首先，我们认为（或者说意识到）博客应该便于浏览者扫描页面，以决定某篇日志是否值得阅读。

报纸上的文章在最开始的几句话中都会展露出高浓度的信息，以此告诉阅读者这是一个关于什么的故事。然而博客并不都有新闻记载那种特色，因此它们很少提供能概括全文大意的开篇陈述。但当访问某个博客时，这正是我们最先想看到的。

其次，我们认为博客应当能更好地鼓励交流。绝大多数博客都为用户提供了添加评论的功能，但我们相信除了简单的日志评论之外，还能够激发更多地交流。

最显而易见的办法就是提供 Trackback（引用通告）[①]。正如你可能已经知道的，Trackback 就是在日志页面中出现的某些摘录，来自于与该日志相关的其他网站。但问题是只有很少一部分博客解释了什么是 Trackback 或者它是如何工作的。而如果只有那些知道它的博主才会使用这种技术，那么多网站间无限互联的可能性就变得微乎其微了。

最后，我们认为好的博客设计应该能够诱使人们阅读更多其他的日志。因为访问某个网站阅读某篇博客日志时，我们通常在该网站上停留的时间仅限于阅读那一篇日志。我们不会继续逗留去更多地了解提供这篇日志的网站、作者或组织。

来自听众的建议包括提供链接指向相关日志、最受欢迎日志，以及其他等等日志分类。同时也应该有链接指向前一篇和后一篇日志（如果适当的话）。

另一个解决方案是提供 RSS 的 Feed。而由于大多数人都不知道 RSS 是什么，或者能为自己做什么，所以指向 Feed 的链接应该添加说明并且使用易于理解的文字。

而正如我在 *Designing the Obvious* 中所论述的，RSS 的订阅页面中应当包括 RSS 的简短说明、提供几个 RSS 阅读器程序的链接，并且应该有像样的格式，而不是一整页密密麻麻的 XML。

当然，我们也认为博客中应当包括所有那些常规的内容，例如作者名（如果该博客有多位投稿人）、日志发表日期、指向网站其他部分的链接、搜索栏，以及其他通用元素。

22.1.1　解决方案

我们讨论的最终结果正如图 22-1 所示。

---

① Trackback 是一种网络日志应用工具，它可以让博主知道自己日志的读者中有哪些人撰写了与之相关的文章，实现了网站之间的互相通告。其规范由 Six Apart 在 2000 年制订，并在 Movable Type2.2 中予以实现。这种功能的意义在于，博主可以脱离只能在对方博客中发布评论的方式，而在自己的博客中发表对该日志的评论。而通过 Trackback 的引用通告形式，被评论者的该日志下方将出现该评论的链接，从而在双方彼此的日志中都能得到体现。这种评论方式使得评论的发表者既能通知被评论者，又能随时维护自己的评论并且随时得知其他浏览者如何看待自己的评论，同时还能通过通告吸引更多的浏览者到自己的网站来。如果博客有了 Trackback 功能，那么任何人都可以通过 Trackback 来发表意见和评论，所有的评论都能追根溯源，从而在因特网上相互联接而织成一张大网。——译者注

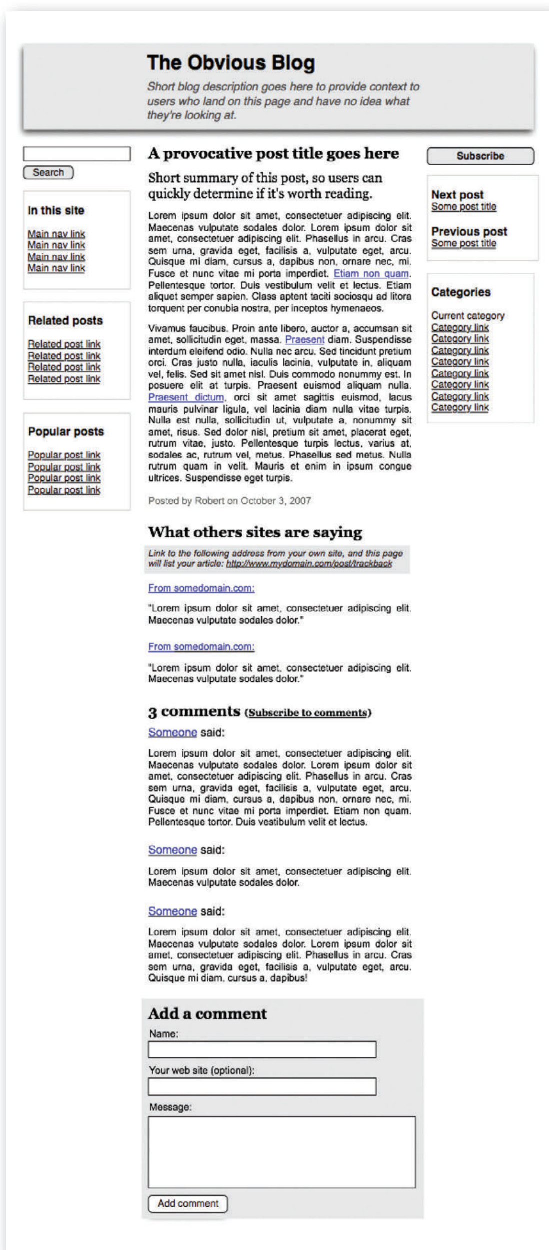

The Obvious Blog
Short blog description goes here to provide context to users who land on this page and have no idea what they're looking at.
Search
In this site
Main nav link
Main nav link
Main nav link
Main nav link
Related posts
Related post link
Related post link
Related post link
Related post link
Popular posts
Popular post link
Popular post link
Popular post link
Popular post link
A provocative post title goes here
Short summary of this post, so users can quickly determine if it's worth reading.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus in arcu. Cras sem urna, gravida eget, facilisis a, vulputate eget, arcu. Quisque mi diam, cursus a, dapibus non, ornare nec, mi. Fusce et nunc vitae mi porta imperdiet. Etiam non quam. Pellentesque tortor. Duis vestibulum velit et lectus. Etiam aliquet semper sapien. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos.
Vivamus faucibus. Proin ante libero, auctor a, accumsan sit amet, sollicitudin eget, massa. Praesent diam. Suspendisse interdum eleifend odio. Nulla nec arcu. Sed tincidunt pretium orci. Cras justo nulla, iaculis lacinia, vulputate in, aliquam vel, felis. Sed sit amet nisl. Duis commodo nonummy est. In posuere elit at turpis. Praesent euismod aliquam nulla. Praesent dictum, orci sit amet sagittis euismod, lacus mauris pulvinar ligula, vel lacinia diam nulla vitae turpis. Nulla est nulla, sollicitudin ut, vulputate a, nonummy sit amet, risus. Sed dolor nisl, pretium sit amet, placerat eget, rutrum vitae, justo. Pellentesque turpis lectus, varius at, sodales ac, rutrum vel, metus. Phasellus sed metus. Nulla rutrum quam in velit. Mauris et enim in ipsum congue ultrices. Suspendisse eget turpis.
Posted by Robert on October 3, 2007
What others sites are saying
Link to the following address from your own site, and this page will list your article: http://www.mydomain.com/post/trackback
From somedomain.com:
"Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor."
From somedomain.com:
"Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor."
3 comments (Subscribe to comments)
Someone said:
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus in arcu. Cras sem urna, gravida eget, facilisis a, vulputate eget, arcu. Quisque mi diam, cursus a, dapibus non, ornare nec, mi. Fusce et nunc vitae mi porta imperdiet. Etiam non quam. Pellentesque tortor. Duis vestibulum velit et lectus.
Someone said:
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor.
Someone said:
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus in arcu. Cras sem urna, gravida eget, facilisis a, vulputate eget, arcu. Quisque mi diam, cursus a, dapibus!
Add a comment
Name:
Your web site (optional):
Message:
Add comment
Subscribe
Next post
Some post title
Previous post
Some post title
Categories
Current category
Category link
Category link
Category link
Category link
Category link
Category link
Category link
Category link

图　22-1

见图 22-2，在页面顶部有清晰的网站标题，紧随其后的是一段概要。这样可以让用户很快地洞察这个站点是关于什么方面的，并且能预料到从中将会得到什么。



图　22-2

它的下方是日志的标题。标题使用了很大的 Serif（衬线）字体以区别于日志正文所使用的 Sans Serif（无衬线）小字体[①]。（这是报纸和网站中常用的伎俩，用于在视觉上将标题和正文区分开来。）

因为日志的标题通常不会展现出内文的主题（事实上越是怪异离奇的标题才越有意思），我们在标题的下方放上了一段简短的内容概要。现在用户可以通过一段简单的摘录决定这篇日志是否值得读下去。

然后，毫无疑问就是日志本身，而其后显示了作者的名字和发布日期。

下一个部分的标题是“What Other Sites Are Saying”，见图 22-3。这个部分显示了 Trackback 的摘录，除了通过某种方式向用户解释了这些摘录是从哪里来以外，同时还说明了如何创建 Trackback，以及它的工作原理。它只是这么一句话：

在你自己的站点链接以下地址，这个页面就会列出你的文章：http://www.mydomain.com/post/trackback”。

① 在西方国家罗马字母的字体分为两大种类：Sans Serif 和 Serif。Serif 字体在字的笔画开始及结束的地方有额外的装饰，而且笔画的粗细会因横直的不同而有区别。相反的，Sans Serif 字体则没有这些额外的装饰，笔画粗细大致差不多。常见的 Times New Roman 等就属于 Serif 字体，而 Arial、Tahoma 等则属于 Sans Serif 字体。中文字体也同样如此，宋体、细明体（繁体中常用）等属于 Serif，而黑体、幼圆等则属于 Sans Serif。——译者注

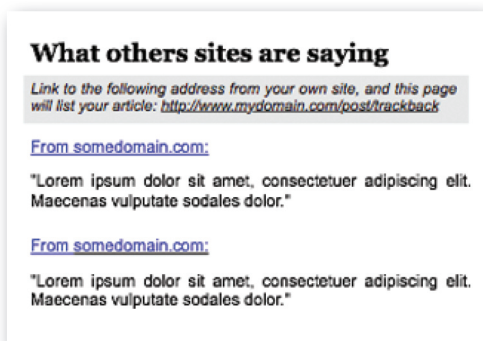


图　22-3

有了这段说明，用户就能很容易地知道在自己网站写的日志是如何在当前页面显示摘录的。让事情变得简单将能鼓励用户去尝试，这不仅鼓励了多网站之间的互联（比如说，某人写了一篇博客日志回应你的日志），同时还增加了你自己网站的访问量，因为在其他网站阅读日志的人有可能会点击 Trackback 过来看你的文章。

同样地，每一个 Trackback 的摘录都会指向其原文所在的站点，因此用户可以点击去看那些日志，从而能够对主题了解得更多，或者看看其他人的看法。

再后面就是评论部分，见图 22-4。标题中显示了评论的数量。这个地方同时还提供了一个链接以便订阅所有的评论（很常见的情况，在对某篇日志发表评论之后，我们希望能看到其他人的反应，但总是忘记回来查看。而如果用 RSS 订阅了评论，任何时候打开 RSS 阅读器你都能获得最新的评论）。

最后是一个简短的表单，让用户发表评论。人们之间的交流只需一点必需的信息——你的名字和网址（选填项）。你的名字将会显示在你的评论上方，而且它还能指向你的个人网站，如果你有的话。这意味着写点评论就能够增加你自己网站的访问量，这也更加鼓励了人们交流。

让表单尽可能地短小简练将有助于人们主动地发表评论。如果这件事很简单又很有价值，那么我们就会去做。而且短小的表单将使这个决定更加容易。

3 comments (Subscribe to comments)

Someone said:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus in arcu. Cras sem urna, gravida eget, facilisis a, vulputate eget, arcu. Quisque mi diam, cursus a, dapibus non, ornare nec, mi. Fusce et nunc vitae mi porta imperdiet. Etiam non quam. Pellentesque tortor. Duis vestibulum velit et lectus.

Someone said:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor.

Someone said:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas vulputate sodales dolor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus in arcu. Cras sem urna, gravida eget, facilisis a, vulputate eget, arcu. Quisque mi diam, cursus a, dapibus!

Add a comment

Name:

Your web site (optional):

Message:

Add comment

图　22-4

左侧的工具栏中有一个搜索栏，如图 22-5 所示。其后是几个链接模块，分别指向网站其他栏目、相关的日志，以及最受欢迎日志。这些链接列表将鼓励用户阅读网站中的其他日志。而更进一步，页面右侧的模块提供了指向其他日志分类的链接，见图 22-6。

右侧工具栏还提供了指向前一篇和后一篇日志的链接，每一个都被清晰地标示出来。

最后，右侧工具栏的顶部还提供了一个大的 Subscribe（订阅）按钮。通俗易懂的术语和便于点击的按钮将会鼓励用户订阅 RSS Feed，至少也能通过 RSS 订阅页面上的信息进一步地了解 RSS。



图　22-5

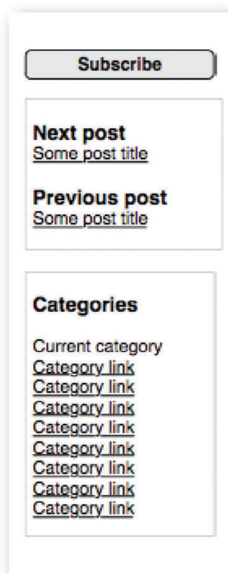


图　22-6

22.1.2　别跟着傻瓜上路

令人悲哀的是，到写这一章为止，连我自己的博客都还没能做出相应的修改。我能有什么资格告诉你们这么做才是对的呢？不过，尽管在开始设计这个博客模板时我没有使用任何设计工具，但仍然有相当令人称奇的地方：300 个博客的狂热分子。

而本章所讨论的设计都是他们认为应该做到的。

合作化和职业化的博客行为仍然是一个相当新的想法，但是用诚恳、真实的声音与客户对话，将能快速地增加你的吸引力。这是接近受众想法和观念最好的办法之一，而且能够提高宝贵的用户忠诚度，就像 Apple、Google 和 37Signals 那些公司一样。

为自己的博客设计多付出一些努力，那么你和用户间沟通的能力将有显著的不同。通过 Trackback 和评论来鼓励交流、为博客日志撰写概要、为相关内容提供链接，这些都是很保险的方法，不仅能改善你的博客，同时也会为你带来忠诚的用户。

在接下来几天里，我希望能有时间把这些点子应用到自己的博客里面去。

第 23 章
邀请讨论

本章我的观点并不是出自于某一个项目。实际上，它们来自于我曾从事的与邀请用户添加回复、撰写评论或者在论坛中参加讨论相关的几乎每一个项目。

每当经手类似的项目，我总是会被问到同样的问题。差不多就好像这样：

> “我们正在考虑是否应该对评论加以控制。你对这件事看法如何？”

也许你已经猜到了，隐藏在这个问题背后的驱动因素其实就是恐惧。

对于由用户自行发表的评论，他们恐惧有人可能会说一些对网站不利的坏话，害怕有人会冒犯其他用户，还担心自己的员工可能会忘记检查评论而没有确保让每个人都玩得开心。

而在这些背后，他们又恐惧自己的员工将不得不把每天的时间都浪费在可笑的看管和删除评论上。

下次如果有人再问我这个问题，我准备让他们看看这一章，因为我一直秉承如下的观点。

23.1 让你的顾客发表意见

一句话，不要控制任何东西。

不要控制留言、评论、论坛回复，或者其他相关的任何东西。为什么呢？因为控制将会带来严重的负面效果。

它会让客户觉得你在隐瞒着什么东西。

比如说，为某个产品写评论时，我们相信自己是以某种方式在帮助其他可能的消费者决定是否应该购买这个产品。否则我们为什么要写评论？

因此，如果我们想说的是好话，就表示我们很高兴，欣然愿意为这个产品做宣传，而这也正是生产该产品的公司所希望的。很明显，这对那家公司来说是件好事。

但是，如果我们有些负面的话想说，表示正在试图用自己的遭遇来警告其他用户。此举并不总是带有恶意。最常见的，我们只是想把使用中遇到的问题和困难说出来，让其他人能事先知道，而不会因为遇到同样的事情而惊讶。是的，有些时候我们也会想要挥着小红旗指挥人们尽可能地远离某个产品，但总的来说，绝大部分产品还没有差劲到如此地步。

如果我们写了这种警告性的文章，然后被告知这些评论需要先通过审查才能发表出来，我们就很有可能会对该网站丧失一部分信任感。不管是有意还是无意，我们都会猜测是否文章在发表之前就会被强大的审查者枪毙掉。而且说实话，因为忙碌的生活，我们并不一定很想回到网站去检查自己的评论是否能发表。如果当时不能看到结果，我们的兴趣也就消失了。（当然，如果我们真的有很难听的话想说，可能会强制自己回来检查，以便确定那些讨厌的审查者是否通过了评论。）

而除了我们因为种种怀疑而感到受了胁迫之外，生产该产品的公司几乎什么都没得到。

事实上，员工们每天花费宝贵的数分钟或数小时来管理这些评论，而公司实际上在付钱抛洒自己消费者的信任。

简直可笑，不是吗？

23.1.1　一个关于信任的问题

有一些很好的理由可以让你完全不再思考评论的问题。

首先，让你的顾客自由发表意见以表示你信任他们的谈论。他们称赞时，你放手让他们去宣传；他们说了不好听的话时，你也相信他们的看法有一定意义。而正如我们都知道的，信任才是建立所有成功关系的基石。

其次，这表示你信任用户自身的判断能力。不去防范每一条负面评论或者论坛争论，你相信用户会运用他们自身的才智、常识和经验过滤纯负

面的内容，在字里行间寻求信息，然后得出结论。

换句话说，给用户自己做判断的权利，而他们自然会忽略那些无用的评论。

如果这些理由还不够，你还可以通过这种方式让那些未经过滤的声音为自己所用，也就是招募自己的 Otaku。

23.1.2　起用你的 Otaku

正如 Wikipedia（维基百科）所描述的，Otaku 是一个术语，用于形容那些“对某些领域有着痴迷热情的狂热者”[①]（这个术语本来主要用于形容热衷于日本动漫的人，然而自从营销大师 Seth Godin[②]在一次演讲中提到这个词后就被广泛引用开来）。

那些抱怨声最大的人、欢呼得最热烈的人、当有人需要解答问题时尽自己最大帮助的人，还有总是在你身边告诉你他们正在想什么的人，这些人就是你的 Otaku。而你可以让他们帮助你。

让顾客们公开地抱怨你的产品、在论坛里表达他们的沮丧，或者征讨公司发表的言论。通过这些你便拥有发现人们真实想法的能力。而任何一个不受约束的看法都比黄金还值钱。

首先，满怀激情地憎恶也比毫不关心的冷漠要好得多。那些最恨你的人，在你真正开始为他们考虑的那一刻，往往会转变成最坚定的拥护者。如果你能做到这一点，那么就能少一个曾经的叫嚣者，而多一个终生的朋友。

其次，你还可以通过组织手段重塑顾客体验。比如说，你可以联系那些喜欢到处宣扬的不安份分子，表示你非常感谢他们提供的反馈以及专业意见，然后邀请他们加入志愿者团队，在论坛里帮助其他用户回答问题。

---

① Otaku（おたく）一词原指动漫与游戏的爱好者，即“御宅族”。现今该词也可用于指称其他方面的狂热者，或对某种领域具有较深造诣的人（尤其是较特殊的领域），如“军事御宅”等。在中国网络流行用语中有许多运用“宅”字自创的词汇，如“宅人”、“宅男（女）”，不过多半已经与御宅族的本质无关而沦为戏谑他人的蔑称。——译者注

② Seth Godin（赛斯 • 高汀）是 20 世纪 90 年代后期以来一系列商业领域专著的作者，也是一位演讲家。其第一部畅销书是关于特许营销（permission marketing）的著作。——译者注

作为交换条件，你可以为他们签署同意书支付一些酬劳，给他们与内部员工直接交流的机会，赠送 T 恤或棒球帽之类的免费礼品，甚至造势让他们成为网站里的交际明星。

举个例子，Adobe 公司推广过一项名为 Adobe Community Experts（Adobe 社交专家，简称 ACE）的计划[①]，目的是为了奖励积极支持 Adobe 产品的社区活跃分子。（这些人不仅仅只是参与论坛讨论，他们可能还发表过文章、出版过书籍或者做过以 Adobe 为主题的演讲等等。）入选者的个人简介将会在 Adobe.com 上公布，同时还得到特制的证书图片，以便能在个人网站上发布，证明自己获得了 Adobe 公司高度的信赖和重视。这些入选者还将受到 Adobe 职员的款待，得到很多内幕消息，并且能与 Adobe 开发团队直接地沟通。

Adobe 公司并没有为这些人支付酬劳。ACE 的成员都是专家级的志愿者。从这方面来看，他们可能也是 Adobe 最好的一个拥护者群体了。ACE 写文章，在会议或者用户集会上讲话，还在论坛里帮人们解决问题。

而这件事的背后是 Adobe 为自己组建了一个预备役。这些人整天挂在论坛上回答问题、提供建议并给其他用户做些推荐。有些人可能认为不管理留言表示公司对论坛不够重视，而其他用户也不会理睬这些留言。但有 Otaku 团队在，他们会解决问题。他们活跃在 Adobe 的论坛以及世界各地的用户组讨论列表中，他们是 Adobe 的客户服务第一线的重要成员。Adobe 将通过 ACE 获知负面评论，同时也能看到高度赞扬的评论。

设想一下。如果你的公司也有这样一群人在宣扬你的 Web 应用，同时在社区里帮助你解决问题，那会是怎样一种感觉？而且，当这些专家们抱怨时，你将知道那肯定不会是无稽之谈；而当他们吹捧你的产品时，肯定也会有很多人相信他们的选择。他们其实是在为你做营销。

---

① ACE 计划的对象是积极贡献和分享 Adobe 产品使用经验及知识的 Adobe 消费者。具体来说，是那些积极在 Adobe 论坛帮助其他用户，在各种在线社区提供 Adobe 技术支持或信息，在专业出版物中发表文章，以及在会议和用户组中进行关于 Adobe 产品演讲的各种用户。该计划网址为 http://www.adobe.com/communities/experts/。

——译者注

23.1.3 标记那些来犯者

最后，为了消除你可能的最后一点担心，我们也可以通过“无痛”的控制方式来添加一点功能，提升一些安全系数：标记系统。

在每一篇留言、回复、论坛帖子、日志评论旁边加一个小小的“标记”图标。这些图标可以点击，用户能对某篇评论进行标记，通知公司员工对其检查。

通过这种方式，当某个无赖闯到你的论坛里时，其他人就可以告诉你他们受到了打扰，让你处理。

这个“标记”图标将用户引领到一个简单的表单，他们可以写下为什么这条评论应该检查。不多不少，保持这种简单。而且，别要求你不需要的任何东西（例如用户的电子邮件地址，因为当用户登录时你已经得到了）。

23.1.4 为他们让路

以上，我的朋友们，当有人问我如何看待控制评论时，我回答的就是这些内容。一字不差。

你不需要控制任何东西。

打开闸门，让你的用户自由地发表意见。为他们让路。

只需一点点的谦卑和一点点的驾驭，你就能把自己的大部分顾客转变成忠诚而又活跃的追随者和拥护者。你可以把自己的恐惧转变为一种工具，以它作为驱动力来收集顾客提供的信息、发展 Otaku，提供出众的客户服务，而且在你的公司和用户之间建立起一种相互信任的文化。

而这些为你赢得的回报，将会远远超过你去控制它们的所得。

第 24 章
得到一个好的评分

从 Amazon 到 Yahoo 再到 Yelp[①]，五星评级系统随处可见，但是各自执行起来却常常有所差异。

比如说，它们经常会分成两个部分，并显示在页面的不同区域。其中一部分是**展示**版本，它告诉用户现在已有的评定结果。另一部分则是**编辑**版本，用户可以在那里选择自己对某事物的评分，例如书籍、唱片或者某篇文章。

在这些评级系统中，其中有些还会在鼠标悬停到星星上时显示说明性文本，告诉用户每颗星代表的等级。有些系统需要先点击某颗星然后再点 Save 按钮，有些则只需要点击某颗星。

但当你只需要点击某颗星时，除了星星从白色变成黄色（或者其他颜色）之外，你无法得到系统的回馈，也就是说你感觉不到交互是否已经完成。这种设计让用户（包括我在内）无法对自己的操作建立信心。

这些都是每当我要设计五星评级系统时所考虑的问题。

甚至某些情况下，由于页面空间有限，我还需要把这两个部分合二为一。我需要通过某种方式使这个单独的界面控件里既能显示当前的等级情况，又能让用户自己评级。

下面将展示我现在是如何做的，然后再告诉你们以前我为之犯过的错误。

---

① Yelp 是一个关于旧金山当地商业机构的用户评述汇总网站，诞生于 2004 年 10 月。用户在 Yelp 上给当地的各种店铺打分留评语，在论坛里讨论，也能通过 Yelp 在本地找到衣食住行的各种信息。网址为 http://www.yelp.com/。——译者注

24.1 清晰胜于效率

为了把展示界面和编辑界面揉合到一起，我把它们分成了两个部分：一部分负责显示，另一部分负责交互。最开始是单纯用于展示的五颗星，右侧则是启动评分行为的链接，如图 24-1 所示。

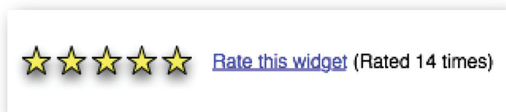


图 24-1

为了让用户能选择自己的评级，在他们单击 Rate this widget（为此评分）链接后那组星星将进入可编辑状态，并且即由黄变白。

当然，用户也需要保存评分结果并回到展示状态，因此 Rate this widget 链接被替换成了 Save（保存）和 Cancel（取消）选项，这也向用户声明现在可以编辑了。见图 24-2。

用户单击 Rate this widget 链接，然后点击他们认为合适的星星来评定等级。图 24-3 中，用户的评分为 3 颗星。



图 24-2



图 24-3

在这里，他们可以单击 Save、Cancel 或者另一颗星。让我们假设现在该用户选择保存自己的评分（同时他更改评分为四星，如图 24-4 所示）。



图 24-4

保存之后，系统将计算最新的等级平均值，然后再次回到展示状态显示新的评定结果。连同 Saved!（已保存！）的文本一起，这将通知用户交互已经完成，同时他的评分已记录在案。在展示状态下他们也能更改评

分，见图 24-5 右侧的 Change your rating（修改你的评分）链接。



图 24-5

我在 Rate this widget 链接后面还加了一点文本，显示已有多少位用户进行了评分：Rated 15 times（共有 15 次评分）（这里的 15 是动态取值）。这不仅能够告诉当前用户自己所做评级在系统中所占的百分比，同时还向他们证实自己的评级结果已添加（很明显之前的数字应该是 14）。

最后，在交互中加入了一些逻辑判断，以防止某位用户为单个对象多次评分。也就是说，如果他在日后又评选了另一个等级，平均值将会更新，但是评分者的个数不会改变。

将展示和编辑版本合为一个评级系统，有好处也有坏处。

从好处来说，它占的位置比两部分分开的版本要少，因为你只需在页面上找一小处地方来放置它就行了。

而不好的地方在于，这种设计中完成交互所需的步骤比其他设计要多。但关键是这种设计中交互的每一个步骤都非常清楚、直接。

就像刚才所说的，两部分的版本可以简单到只需要点击某颗星而已，但是系统无法提供反馈，你无法知道自己的评分是否被保存，或者是否有效。将两个功能结合在一起意味着你的每一步都能得到一个清晰的结果。

单击 Rate this widget，界面将进入可编辑状态。点击某颗星将把它变成黄色。单击 Save 让你知道自己的评级结果已被保存，等级平均值和评级总人数都将会更新。

现在的交互确实需要更多的步骤，但是它做得更好的地方在于提供了适当的反馈，所以用户能在与该系统交互的每一个瞬间都保持自信，并且每一步都很清晰。

换句话说，效率并不一定是所有问题的答案。我们在第 16 章看过那个保险公司的案例，最短的并不一定就是最好的。

比效率更重要的是清晰。对于那些第一次来到网站，或者执行自己不太熟悉的任务的用户来说，最好是让每一个瞬间都尽可能地易于理解。对于那些对使用产品很有经验，或者执行非常普通的任务的用户来说，则最好是尽量缩短流程，别让过程变得让人厌烦。不管是哪种情况，最关键的是让每个步骤都清晰易懂，同时不太复杂。

用户一般不会频繁地和评级系统打交道，而且还得面对不同网站的各种版本。因此我宁可稳妥一点：只求清晰，不求效率。

信任该信任的

我曾经使用过上述设计，但是其中没有 Rated 15 times 的说明，那只是由于我当时还没有想到。那时，我认为自己已经得到了一个足够棒的设计，并且在一次关于 *Designing the Obvious* 的演讲中谈到了它。

当时有一位听众建议说这个系统里面应该声明已经有多少人进行了评分，这能帮助用户更好地判断自己的评级是否有意义。毕竟，如果只有一个人做了评价，哪怕是五星，也并不一定就表示这个产品很伟大。它只表示现在有一个人认为这个产品很伟大而已。

而如果有 30 个人给它评四星半，相比之下则会有意义得多。

我站在那里，心里想，这真是个好点子。

我告诉那位可敬的先生我将剽窃他的想法，并且会用到我的下一本书里。

所以，我在此专门提及此事。谢了，伙计。

第六部分

管 理 信 息

几乎每一天，我都要花上两三个小时来处理电子邮件、浏览新闻、阅读博客日志，然后再查看那些 Blog 里所有我感兴趣的链接。这还算是好的，有时候根本不止两三小时。

毫无疑问，我是想一下子管理太多的信息，但这个话题我们可以换个时间来讨论（还有，我在给客户账单时可从没把这些时间也算进去）。

我想说的是，网上的信息量极其巨大，而管理它们可是一项绝不轻松的工作。我们每天为管理这些数据分类所花费的时间都只是为了达到一个目的：组织它们。这样我们才会觉得自己是生活积极的参与者，而不是愚蠢地让珍贵的时间匆匆流逝。

对于那些访问我们网站或者使用我们产品的人来说也是一样，他们在网上需要学习新知识，需要做计划，以及其他各种各样的事情。不管有没有意识到，他们一直都在乞求用科学的方法有效地获得每天最想要的信息。而尽管现在有很多相关的技术，但是大部分仍然未能进入主流。

即使是 RSS 这种从可信赖站点获取新闻及行业见解的解决方案，也仍然没能成为一个能被广泛理解的典型范例，更别说成为家喻户晓的词汇了。

因此，在接下来的几章中，我们将针对如何推动信息管理加以阐述。我们将讨论如何把诸如 RSS 和 Tag（自定义标签）模式等陌生的想法阐述得更有实际意义、如何通过拖放操作来组织页面内容，以及如何使用系统通知来管理中断。

我们甚至还会讨论为*改变而设计*的概念，这样用户就能在一头掉进完全崭新的设计而晕头转向之前先适应你的想法。（我们已经对此考虑得足够多了，一点不需要夸张！）

这些瞬间都应该表现得非常简单，并且我们有几种途径可以达到这个目标。

第 25 章
让 RSS 更有意义

要想让用户战胜对新生 Web 技术的陌生感并加以运用，最困难的一点就是首先要意识到人们并不关心技术本身，而是关心自己能得到什么。我们推广的并不是技术，而是技术所能带来的好处。

为了说得更明白，我们来看看 RSS。RSS 由于各种原因并没有获得广泛认可，造成的原因有很多。比如，学习新事物有潜在的困难。再比如，使用 News Reader（新闻阅读器）所带来的好处也许并不值得我去安装，并让它进入自己的正常生活。而更重要的原因是，事实上绝大部分人甚至包括一些已经使用 *News Reader* 的人根本就不知道 RSS 这个词到底是什么意思。这几个缩写字母常常被用在某个按钮上，指向订阅页面。但因为不理解 RSS 的意思，用户也很少会去单击这些按钮。毕竟，按钮标签应该在用户脑海中建立清晰的预期，并且告诉他们点击后将会发生什么。

推广某项新技术时，我们的首要目标应该是让它通俗易懂。人们愿意学习新的想法，但这些想法必须能够学习。这一点对于任何新技术都是适用的。

有些网站在启用 RSS 时已经开始尝试解决这一问题。

FeedBurner.com[①]在很早以前就告诉我订阅 RSS 并不像访问牙医那么可怕。他们并没有显示丑陋、冰冷、满是 XML 的页面，然后等着用户自动明白原来需要把 URL 复制粘贴到阅读器里面去。FeedBurner 在这里表现得更为友好，如图 25-1 所示。

---

① FeedBurner.com 致力于通过博客和 RSS Feed 的形式在网络媒体和广大受众之间进行联接。网址为 https://www.feedburner.com/fb/a/home。——译者注

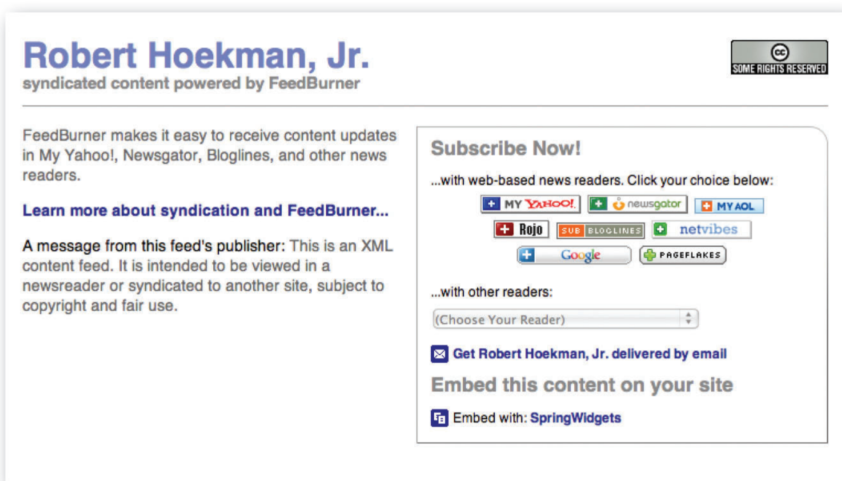


图　25-1

他们展现在我面前的是一个相当吸引人的页面，有着整齐的博客内容。而在越来越多的在线和桌面 RSS Reader 程序冒出来的同时，FeedBurner 提供了一种被称为 Chicklet[①]的东西。它只需点击一次就能轻而易举地订阅 Feed。该页面中没有任何 XML，而且 FeedBurner 也避免使用过多的专业术语。RSS 缩写根本没有在页面中出现。

谢谢了，FeedBurner，不胜感激。

如图 25-2 所示，AddThis.com[②]所做的也比较类似。他们在页面上丢了一堆 Chicklet，这样任何使用过 BlogReader 的人都能很容易地订阅。AddThis 同样在设计中完全摈弃了 RSS 的缩写，而是使用 Feed Reader。

这些网站在将新点子通俗化方面成效斐然。但是同时它们也排斥了一些东西。这些网站没有提供其他可以订阅 Feed 的方法，事实上它们只关注通过 RSS Reader 来订阅。

---

① Chicklet 也被称为“社会性书签”，实际上就是一种小图标，以图形化的方式声明 Reader 的出处，用户只需点击该图标即能通过所选择的 Reader 来进行订阅。
——译者注

② 和 FeedRurner 一样，AddThis 同样是最热门的 Web2.0 网站之一。它可以说是网上书签和共享按钮的标准制定者。网址为 http://addthis.com/。——译者注

其实还有很多其他选择。我决定深入探索，看看如何能把其他方法也清楚地展示出来，这样用户就能轻松地理解并且运用它们，而且这样做的好处是显而易见的。

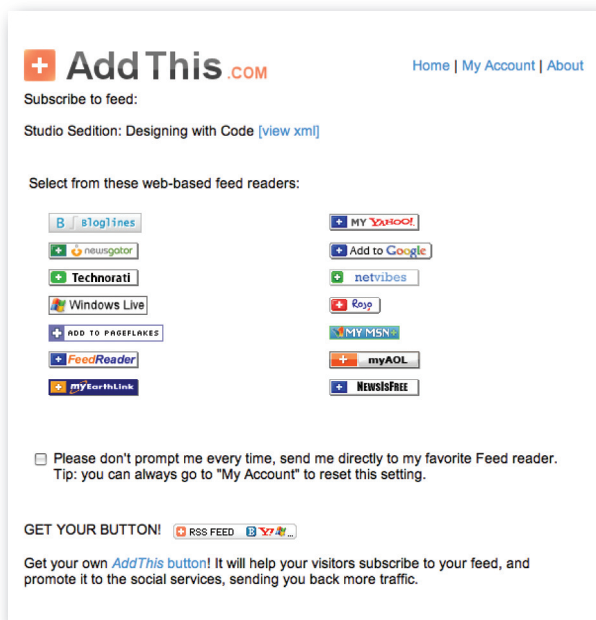


图 25-2

25.1 破译其他选择

首先我在通往订阅页面的按钮上使用了 Subscribe（订阅）的标签，而不是 RSS。使用能够建立强烈直观印象的词汇可以消除语言障碍，并且降低对新生事物的学习曲线。这很简单。

然后我借鉴了 FeedBurner 的方法，在页面上创建了一组 Chicklet。这里我仍然使用了在第 22 章讨论过的博客模板，如图 25-3 所示。

这里的想法是向用户展现一个看上去仍属于本网站的页面，而不是把他们送到类似 FeedBurner 或者 AddThis 那样的页面中去。因此我保留了工具栏中的站点导航区域以及搜索区域。

那些已经拥有各种在线或桌面 Reader 程序的人，只需单击某个 Chicklet

就能完成订阅。



图　25-3

有些人所用的 Reader 没有在页面中列举出来，我也为他们提供了订阅的方式，并在旁边附加了说明。他们所要做的只是简单地把 Feed 地址复制并粘贴到自己的 Feed Reader 中去。文本框中，网址已经默认选中，而旁边就是 Copy（复制）按钮。通过这种方式，用户只需单击 Copy 就完成了在该页面的工作。

以上是订阅 Feed 的第一种方法，所以我把注意力放在了其他选择上。

首先是我添加了一个选项以便收取关于博客更新的电子邮件通知。见图 25-4 下方。

有些网站，包括 FeedBurner，允许用户把 RSS 转化为电子邮件的 Newsletter（时事通讯）订阅服务。因为在今天这是一件很容易的事情，所以我们无需任何努力就能添加这一功能。实际上，提供这种通过电子邮件获取内容更新的选择将会极大地增加你的订阅率。因为电子邮件仍然是网络营销最有效的方式之一（并不是每个人都阅读博客日志，即使他们拥有某个 Reader 账户。更多的人还是愿意阅读他们的电子邮件）。



图 25-4

进行这些设计时，我在标题上还加上了 Option 1（选择 1）和 Option 2（选择 2）的字眼。这样用户就能通过推理很快地认识到第一部分并不是他们唯一的选择。

然后最后一种方式是通过浏览器订阅并浏览 RSS Feed。我添加了第三个部分，见图 25-5。

末尾这一部分包括了一段指示性文字，说明如何使用主流浏览器（IE、苹果的 Safari，以及 Firefox）来订阅 RSS Feed。

最后，为了进一步让该技术更加容易学习，我把术语 RSS 全部都换成了 News Reader。

清晰的标题、简单的指示说明，以及简便、快速的解决方案已经迈出了一大步，让订阅流程变得更易于理解，并且值得尝试。去掉 RSS 缩写只是为了让该技术更易于接受所做的最后一点尝试。

使用 News Reader、订阅电子邮件通知、通过浏览器本身来管理 Feed。用户现在有了多种选择来订阅内容，而且每种方式都无需付出过多努力。而把讨厌的专业词汇剔除出去，即使是那些业界之外的人也能很容易地理解这项技术。

Search

In this site

Main nav link
Main nav link
Main nav link
Main nav link

Subscribe to "The Obvious Blog"

Back to the blog

Option 1: Use a news reader

If you already use one of the following news feed readers, click it to auto-subscribe to this blog. If not, clicking the button will take you to a registration page.

netvibes　Bloglines　newsgator
myAOL　Google　Rojo
MY YAHOO!　Pageflakes

You can also simply copy the web address below and paste it into the Add Feed (or similar) field in your news reader.

http://feed.mydomain.com　Copy

Option 2: Receive email notifications

Enter your email address below and we'll send you an email whenever there is new content available on the blog.

e.g. me@mydomain.com　Subscribe

Option 3: Use your browser

Most modern browsers offer a way to subscribe to content so it can be read in your browser instead of through a news reader. Following are instructions for how to do this in several browsers.

Internet Explorer 7 (Windows only)

[Instructions go here ...]

Safari (Windows and Macintosh)

[Instructions go here ...]

Firefox (Windows and Macintosh)

[Instructions go here ...]

图　25-5

一个新想法是否容易理解是其成败的关键。诸如 RSS 和 XML 这种词语会妨碍用户形成稳固的直观印象，也就很难从中获利。

除掉它们，事情就会让人很容易接受。

第 26 章
自定义标签

在我开始喋喋不休地讲述关于 Tag（自定义标签）的问题之前，你们应该先弄明白一些事情。

我真的很喜欢 Tag 这个想法。为某个单独的内容应用多个 Tag，意味着它将更容易查找，而且这种组织方式更富弹性，远比现代操作系统所用的基于文件夹模式的组织方式要好得多。

通过 Tag 模式，你可以为一封电子邮件定义多个标签，例如 work（工作）、to-do（待办事宜）、this_week（本周）或者 projects（项目）。你也可以把某个资料文件划归为 library（图书馆）、research（研究文档），然后再为它加上具体的标题，例如 usability（可用性）或者 user_expeirence（用户体验）等。

Tag 模式的这种弹性意味着你将能通过无数种方式来查找和组织内容。你可以用任何自己习惯的方式快速地定义它们。

爱死它了。

但是 Tag 本身并不是一个完美的系统。如果你为社区用户提供了 Tag 模式，有时候事情会变得很糟糕，因为这里面有许多潜在的问题。

首先，就像 RSS 一样，Tag 模式也是一个全新的想法，但并没有多少网站能让它不言自明，或者阐述得很清楚。其次，地球上没有哪种语言是用下划线来分隔字词的，所以 Tag 用到的术语和语法所带来的不熟悉感将会在使用时带来混淆。第三，人类并不是机器人，因此同一个 Tag 可能会使用到各种各样的字眼。而最后一点，运用 Tag 模式的网站中所用的搜索引擎对 Tag 标签的支持会远大于对自然关键词的支持。

比如说，我在 Flickr.com 上搜索第一届 Voices That Matter 峰会[①]的图片时，使用的关键词是 Voices That Matter Web Design Conference（Voices That Matter 网页设计峰会）。我用 Google 搜索时就用的这几个关键词，所以在 Flickr 里面我同样用它们来搜索，如图 26-1 所示。

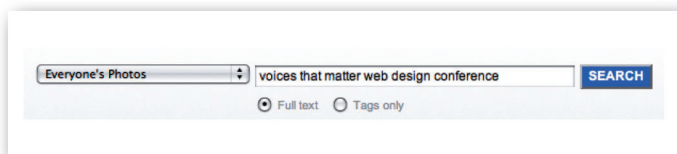


图　26-1

但是 Flickr 只返回了 4 个搜索结果。

我把关键词改为 Voices That Matter，得到了 700 多张图片，但其中有三分之一都和该峰会毫无关系。

我重新在关键词中加入了 Conference（峰会），这次只得到了 58 个结果。这肯定出了问题，因为我知道仅该会议协调人自己就上传了至少 200 张照片。

用了 3 个最明显的搜索关键词，却几乎什么也没得到，于是我决定用 Tag 标签试试。我输入了 VTMWDC，也就是 Voices That Matter Web Design Conference（峰会全称）的缩写，最终得到了 294 个结果，其中每一个都来自于那次会议。还真是个好关键词！

当然，我无法知道这个 Tag 是否涵盖了那次活动的所有图片。而且我也不太明白自己在一开始是怎么想到用这个会议的缩写来作为关键词的。

如果没有想到用缩写字母来搜索，我可能就会被困住。但 Flickr 并不能期望每个人都能自己发现原来用 Tag 搜索才会更有效。而且如果是要查找某些特殊的东西，我又怎能知道应该使用哪些 Tag 来搜索呢？后来我

① Voices That Matter 面向技术、设计和商业界的领袖人物，为他们展示那些“投身于新技术、新方法和新发明”，并“做出了影响业界的杰出贡献”的人。网址为 http://www.voicesthatmatter.com/。——译者注

又使用了 VTM 进行搜索，结果得到了更多的相关图片。

为了让 Tag 系统变得真正有效，还需要再附加一些功能，让创建和使用 Tag 标签能够更简单、更轻松。

26.1 分类系统、分众分类系统以及变异体

有两种不同的方法可以组织网站内容。一个是建立**分类系统**（Taxonomy），也就是说一旦你定义好类别，所有人都将遵循这个分类。另一个是**分众分类系统**（Folksonomy[①]），它算是分类系统的一种协作型式。通过这种方法，用户可以根据自己的需要持续不断地添加和改变分类。

如果你决定运用分众分类系统、使用 Tag 标签，那么上述所有的问题就会随之出现。因而很关键的一点就是如何让 Tag 模式显得更为友好，同时支持用户的思考方式。

所以当我第一次设计自己的 Tag 标签系统时，考虑了以上所有情况，试图寻求一种能改进现有情况的解决方案。

26.1.1 消除语言障碍

为了让初学者更容易理解，我尝试去掉专业词汇 Tag（自定义标签），而是使用 Label（标签），就像 Google 基于网页的电子邮件服务 Gmail 那样，见图 26-2。

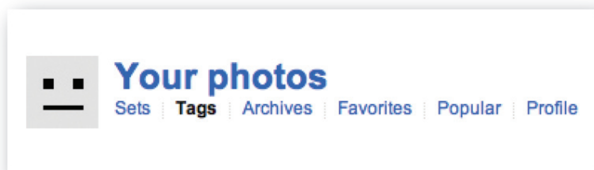


图 26-2

① Folksonomy 是由 Folks（人群）和 Taxonomy（分类法）合成而得的生造词。它由社会性书签服务中最具特色的自定义标签（Tag）功能衍生而来，指"群众"自发性定义的平面非等级标签分类。有人称之为"分众分类"，也有解释为社会分类（Social Classification）。——译者注

“标签”植根于现实世界，因此大多数人都很容易理解。我们无时无刻不在使用标签，有的人在中餐馆的外卖盒上给剩菜做记号，还有的人在装满杂物的糖果罐上贴便笺纸以便于日后翻拣。

而另一方面，“自定义标签”并不是我们日常生活中经常接触的事物。它是个新概念，就像在第 25 章看到的 RSS 缩写那样，使用更加熟悉的词语能够降低对新事物的学习曲线。

为了进一步让 Tag 系统更加友好和熟悉，我认为在创建 Tag 时不应该使用或者显示下划线（例如，my_house）。

尽管系统记录 Tag 的时候需要保留下划线（因为 Tag 将被用于动态产生站点的 URL，而其中不能包含空格），它们也不需要以这种方式展现给用户。总之，它们能够而且应该像正常语言那样显示。

26.1.2　阐述新想法

其次，我需要确保 Tag 模式的目的和用法都表达得足够清楚，所以为此我提供了一些简单的说明文本。换句话说，我尝试不要去假设（如第 7 章中所描述的）用户已经知道 Tag 是什么以及是如何工作的，而是努力提供一些简洁明了的文本来解释它。

我用了一种简单的、类似推荐广告的形式：

> **为它贴上标签**
>
> **添加标签（例如 my house）将会使你更容易地找到需要的图片，并且随意添加你想要的标签。我们将会把标签转换成链接，你只需单击某个链接就能看到所有使用该标签的图片。**

26.1.3　提出建议

不同的人可能（而且几乎总是）会不知不觉地为同一个 Tag 创建多个版本。所以我们需要某种方法来侦测那些相似的 Tag，并且让用户能够为想标记的内容选择那些已有的 Tag。

例如，某个人在一个照片共享网站上传了一些万圣节派对的照片，他可能会把每张图片都标记为 halloween（万圣节）。而参加该派对的另一个女孩则可能会把她上传的照片标记为 Halloween_party（万圣节派对）。两

个 Tag 都很恰当，但是现在针对同一件事却有多个不同的 Tag，因此参加那次聚会的其他人可能无法在一次搜索中找到所有的图片。

（顺便提一句，为了解决这一问题，Yahoo! 曾经在 2006 年的 South by Southwest 交互年会上推行过一种较缺少技术含量的预防措施。在那次 Yahoo! 的官方宴会上，宴厅中张贴着一些广告标语，指示宾客将本次所有的 Flickr 照片都标记为 Yahoo!_bar_tab，这样所有的参加者都能够找到它们。这项举措目前还没有席卷全国，但是想想在每一个宴会、派对、会议或者其他活动上都要告诉大家应该用什么 Tag 来记录活动，这也很有意思。）

我做了一些不同的事来解决这个问题。

首先，我认为网站应该像 Google Suggest 那样提供“自动完成”功能。用户输入某个 Tag 时，系统会搜索出近似的 Tag，然后把这些 Tag 以列表形式呈现，让用户选择其中最适配的一个。

当然，这个办法潜在的不足之处，如同第 9 章所描述的，就是用户有可能会选择一个推荐的 Tag，但其实他自己想输入的 Tag 更合适。

第二种方案是在用户输入了自己的 Tag 之后显示一个推荐页面。用户输入 Tag 之后，系统搜索出近似的 Tag，然后在下一个页面中陈列出来。这个页面可以用文字向用户解释他可以把自己的新 Tag 和那些已被其他人创建出来的 Tag 相匹配。

通过这种方式，为相同 Tag 创建多个版本的几率将会降低，同时查找相关内容也会更容易。

26.1.4　搜索，搜索，再搜索

最后，网站的搜索功能必须为自然关键字提供有效的搜索结果，而不能只针对 Tag（就像 Flickr 那样）。

我知道从开发角度来说这一点也许不太容易做到。但是搜索是大多数网站最重要的一个功能，而且它运转是否良好，这一点至关重要。如果用户不能用 Google 能接受的自然语言找到他们想要的内容，在网站里费力寻找信息将会很痛苦。

26.1.5　自定义标签的未来

再次重申，新生想法应该易于学习。剔除专业词汇能够改善大部分使用 Tag 系统的网站，但很明显，问题不仅仅出在术语上面。

Tag 并不是大多数人都熟悉的东西，因此必须采取措施帮助用户对系统如何工作形成清晰的理解，让他们无需付出多少努力就能搜索到想要的内容，同时用其他用户都能轻易找到的方式组织页面内容。

第 27 章 通过拖放来组织信息

那些帮助你管理信息的产品最大的一个卖点就是组织信息的能力。使用 Dashboard HQ，用户能够创建链接列表和个人注释，并且能完全地控制页面内容。而由于页面内容百分之百都是由用户决定的，因此他们是否能随自己的需要来组织信息非常重要。

为了达到这一目标，我们在产品中提供了创建和删除模块、记录注释以及书签等功能。同时我们还提供了通过拖放（drag-and-drop）操作来重新排列模块这一功能，它能够使用户建立最适合自己的顺序，从而在下次访问页面时能更容易地找到自己想要的东西。

然而，为了让拖放操作顺利进行，这个过程同样需要做到清晰和简练。

27.1 交互的三种状态

一个有效的拖放交互需要提供完整的 3 种状态。

你问这 3 种状态是什么？

27.1.1 “邀请”

首先是“**邀请**”状态。此状态包含页面上任何邀请你采取行动的元素。它可以是触发某个行为（而不是载入某个新页面）的按钮或者是命令链接，或者可以是当鼠标悬停到某个图标上出现的工具提示，甚至可以是一个改变了的鼠标指针，提示用户除了单击之外还可以进行其他操作。

一旦用户在 Dashboard HQ 的页面上单击了 Edit This Page（编辑本页面）的链接，就会进入编辑模式。他将会看到“邀请”状态提供了几项任务可以选择，这几乎也是所有交互行为的默认状态。其中一个选项是删除某个模块，另一个是为模块重命名，而最后一个是通过拖放重新排列模块。

这里的“邀请”就是显示在每个模块左上角的文字 drag（拖曳）。如图 27-1 所示。

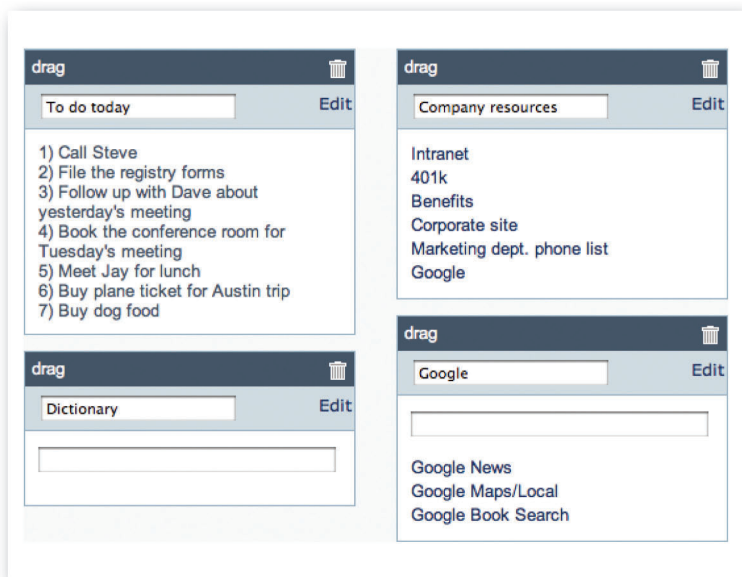


图　27-1

很简单，不是吗？出现它就表示邀请用户采取行动。

为了让“邀请”更加完善，我们设定当用户鼠标移动到文字 drag 上时，将显示代表“移动”的指针，这进一步传达给用户这里能够支持更多的行为。

27.1.2　“操作”

第二个是“操作”状态，包含用户要完成一项任务所必须采取的所有行动。它可以只是一步操作，例如单击某个链接载入新页面；也可能需要多个步骤，例如复制某个 RSS Feed 的 URL 并粘贴到你喜欢的 News Reader 里面，然后保存这个 URL 来订阅该 Feed。

Dashboard HQ 的“操作”状态包括单击 drag 操作柄，拖曳模块到新的位置，然后释放鼠标按钮（或者提起触感笔）让模块“落”到这个新位置上。

每一个步骤都必须尽可能清楚，让用户能很容易地学习并且理解这种交互行为。

稍稍离题一下，不论其面向范围和目的，上述内容对于任何交互都至关重要。所有的交互行为都必须尽可能地清晰。

提到这一点是因为，很多读过 *Designing the Obvious* 的人都建议，我应该把书里面的想法应用到更大型、企业级的应用软件里面去。

我的回答永远都是一样的。设计一款大型的软件当然意味着你需要做更多，而且需要考虑更多，但是它并没有改变通用的规则。

大型软件仍然需要提供最相关和最基本的功能，通过启发式的设计让用户提高使用效率，并且保持从一个界面到下一个界面的一致性和连贯性，同时通过设计元素让交互变得更易学习，并且能重复同种交互行为。而每一个交互行为的每一个步骤都需要尽可能地简单和清楚，特别是在“操作”状态中。

不是简单，也不是清楚，而是简单又清楚（有时候这意味着需要更多简单的步骤，如第 16 章所讨论的，而有时候又意味着需要执行更多的任务，每个任务中包括很少的步骤）。如果有什么区别的话，只能说相较于我经常谈到的小型 Web 应用而言，在大型软件中这些显得更加重要而已。更大型并不意味着更加复杂。更大型只意味着，唔，它更大而已。

现在让我们回到主题。

不管这位用户是第一次操作还是第 400 次操作，拖放交互中的每一个步骤都需要表现其自身意义。

因此，只要用户把鼠标移动到 drag 操作柄上，单击，然后开始拖曳模块，我们就需要确切地传达出因为位置的改变而对页面产生的影响。

为了做到这一点，我们创建了一个虚线框来显示模块将会“落”到的新位置，同时也移动了其他模块的位置，以此确切地表现出每一个元素因此将会受到的影响。如图 27-2 所示。

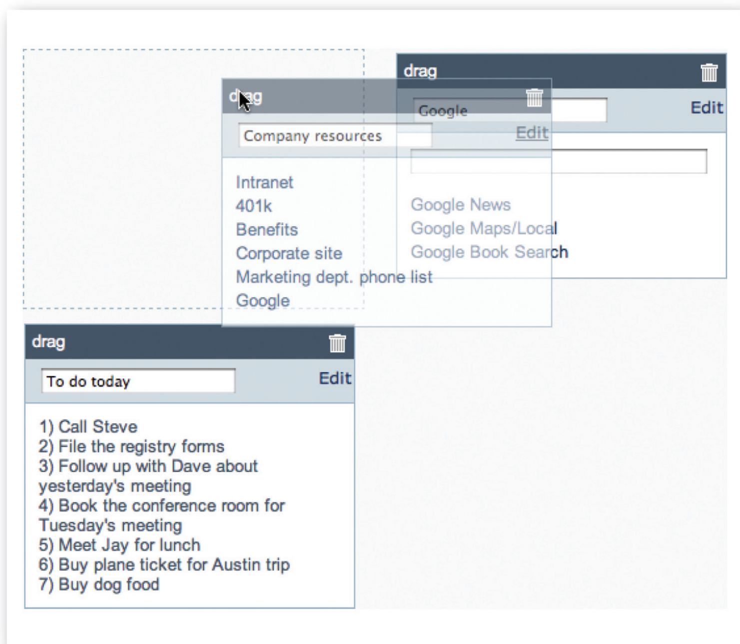


图　27-2

在这里，“操作”状态所需要做的只是拖曳某个模块到新的位置，但在操作的同时，界面也根据需要进行更新，以确切地显示出操作将对页面产生的影响。

27.1.3　“实现”

第三个是“**实现**”状态。几乎每个人都曾经把它忘记过。相信我，我自己就干过这事。但也许是因为“做事就要有始有终”的想法作怪，每当我忘记它时，当天晚上肯定会在床上辗转反侧，苦苦思索为什么自己的设计无法令人满意，所以这对我来说是个教训。现在我已经学会牢牢记住它了，不然等着我的肯定又是一个不眠之夜。

“实现”状态就是让用户知道他的交互已经实现了。这听起来似乎非常简单，但是别忘了，令人不满的交互行为会让人对设计者产生不满。

“实现”有时候只需显示 1 行文字，告诉用户他的个人资料信息已经保存。有时候则意味着当用户在“联系我们”表单中发送电子邮件时向他

们表示感谢。还有的时候它表示在某人从你的购物网站上成功订购后发送的一封确认邮件，以及发货状态的更新。

不过，有时候对“实现”的指示也不需要太明确。在大多数拖放交互的例子中，当被拖曳的模块和其他模块被“吸”到各自新的位置时，就已经充分地表明“实现”了，如图 27-3 所示。

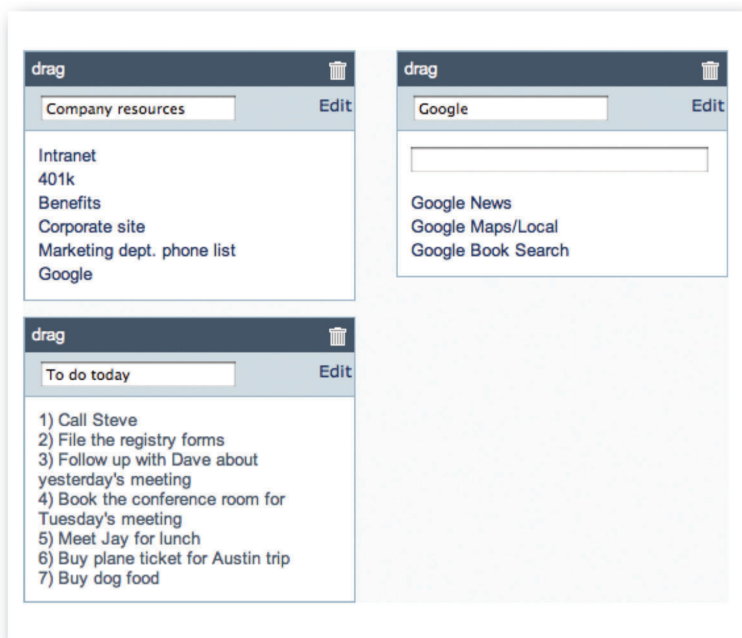


图　27-3

这就是 Dashboard HQ 的做法。别的什么都不需要，而且多了反而会让交互行为更加复杂。

人们那种最普通的、几乎是天生的对于在桌上排布纸张或者拼图所产生的直观印象，这对于拖放操作来说就已经足够了。更别说他们使用计算机时，几乎每天都要用鼠标来回拖放文件了。

27.1.4　完成的感觉

不论是复杂还是简单的交互，我们都需要呈现这 3 种状态，才能让用户理解如何开始、如何完成，然后相信自己操作无误。

我们很容易就会忘记“实现”状态，例如在某个五星评级系统中你点击了某颗星，然后什么也没有发生（如第 24 章所讨论的）。但如果没有它，用户可能就会对自己的操作不太确信，也就没有信心继续其他操作。

不管是简单还是大型的应用程序，请确保每一个步骤都清晰，并且给出适当的反馈。这样所有的交互行为都将会更有效率。

27.1.5　判断你的功能是否正当

拖放操作可以适用于各种地方。在 iGoogle[①]和 My Yahoo![②]上，用户可以在自己的个性化主页上重新排列各种模块。而在 Ning.com[③]上，用户可以把自己想要的元素图标拖曳到自己主页的缩略图里面，并随意安排它们的位置。

但是，尽管设计得非常好，拖放操作和个性化主页这些功能仍然需要视环境而定。当包含它们的这个页面是用户的“目的地”时，它们看上去会很不错；但如果用户只是“路过”这个页面，则没有什么必要。

不同的应用扮演着不同的角色。例如某个项目管理系统，可能一连几天都会在浏览器里保持打开的状态，持续不断地被核查、更新和回顾。它就是“目的地”。你将会在其中花很长时间，因此它可能需要某种合适的个性化定制。

而“路过”则是指你在通向更重要事物中才会使用它，可能还只是偶尔使用。

政府门户网站是个很好的例子。你登录政府门户网站是为了寻找一些确实需要的信息，例如如何在线延长驾照的注册期。像这样的一个页面不需要个性化功能，而且绝大多数情况下，这种功能都将被无情地浪费，甚至会让寻找信息变得更加困难。

---

① Google 为用户提供的个性化主页。用户可以在主页上添加各种个性化工具，例如天气、Gmail、日历、财经要闻等，同时可以自由排列各工具模块的位置。网址为 http://www.google.com/ig。——译者注

② Yahoo! 为用户提供的个性化主页，功能类似 iGoogle。网址为 http://my.yahoo.com/。——译者注

③ Ning.com 是一个社区交互性网站，个性化主页的功能类似 iGoogle 和 My Yahoo!。本书作者在 Ning.com 上的主页地址为 http://designingtheobvious.ning.com/。——译者注

进入这样的页面时，我们很快地扫描，找到第一个看上去合理的链接，然后尽快离开这里。这不是那种我们耗费宝贵时间来根据特殊需求进行定制的页面。

但可笑的是，我经常看见这种“途径”页面变成了极富功能性的个性化主页，仅仅是因为所有者中的某个人认为既然它适合 iGoogle，那么也将会适合他们。

设想一下你的政府网站变成了 3 列，其中满是可以拖曳的模块，你可以查询天气、得到最新的股票行情，甚至还能玩纸牌游戏（这些都是你在 iGoogle 上能做的事情）。

那会很奇怪，不是吗？这些不是人们在政府网站上寻找的东西，而且政府也根本不会因为提供了这种功能而获得任何好处。这种网站应该把注意力放在那些人们在这里真正想找的政府服务信息上面。

另一方面，诸如 iGoogle、My Yahoo! 和 Dashboard HQ 这种服务，它们是由用户设计的，而不是为用户设计的。这些页面里的内容完全由用户决定，就好像他们在设计自己的门户网站——自己的“途径”页面一样。因此，这些看上去是“途径”的应用，变成了“目的”应用。

不知道你会怎么样，反正我不会把时间都花在政府网站上，不管它有多酷。也许那些政府职员会把它当作自己的个人主页，甚至认为个性化功能非常有用，但这些人对于网站的整个受众群体来说，只占非常小的一部分。网站不应该为这些人而设计，而应该为受众中人数最多的群体设计。

而当这种功能确实有意义时（就像 Dashboard HQ 那样），交互是否流畅而愉快就显得至关重要了。确保交互行为的 3 种状态都清晰、明显，将会让用户更加自信。

第 28 章
用系统通知来管理中断

假设你们公司正准备为旗下最受欢迎的 Web 应用进行一次重要更新，同时假设所有的设计和开发工作都已经完成，该应用现在只等待用户的检测，而且你计划在本周内就按下开关发布这个新版本。

然后，假设你在这个紧要关头给我打电话，描述了你们的计划，并且希望我为新的设计做一次可用性审查。

想知道我会说什么吗？

首先，我会说下次你应该更早通知我，但那是另外一回事。

其次，我会问你是否能给自己所有的用户发送一封电子邮件，并且在首页上做一点小改动，然后创建一个页面专门说明即将做出的更新。然后我会问你是否可以在明晚之前搞定这些。

没错，我是个苛刻的工头。

事实上，我见过太多的 Web 应用直接就这样跨过门槛，对那些最拥戴它们的用户却连一句警告也不提。而这些用户正是那些豪爽地买单而且为它们在朋友中做宣传的人。如果连这些用户都不知道即将的改变，更别说那些仍旧在学习并且为你的产品是否值得使用而犹豫的新手了。

其实，告诉顾客你们正在做一些可能会影响他们的事，会带来很多显著的优势。以下是这么做的几个主要原因。

- 它会给你最忠诚的顾客带来一些值得激动和期待的事情。
- 它会给你最保守、最讨厌改变的用户足够的警告，这样他们就能提前对新功能的冲击做好充分准备。

- 它会在你的更新发布之前为人们创造谈资，这样你的新版本将会迎来巨大的访问量，而且口耳相传将会为你带来一些新用户。
- 有了它，用户就不会在某天登录后突然发现所有元素的位置都被移动，或者常用任务流程发生了改变。它会让用户知道自己即将需要一些再适应，而且会很值得。

悲哀的是，很多公司从来都不愿意付出哪怕一丁点努力来收获这些利益。我一直试图让我为之工作或合作的公司进行一些此类的尝试，但是由于各种原因，它们总是疏忽了这一点。

有时候是因为开发团队即将面临项目的最终期限，只是时间不够的缘故。有时候则是因为市场团队不敢事先泄露秘密，害怕在正式发布前被某人剽窃了整个点子（这种观点很糟糕，因为竞争是一件好事，而且“争谁更优秀”远比“争谁更快进入市场”要好得多）。还有些时候仅仅是因为领导层并不认为这事很重要。

你知道，因为良好的客户服务和稳固的营销业绩并不很重要。

荒谬。

我们的顾客和我们自己一样，生活中有大堆的事情需要担心。我们的工作不是某天突然用一个新界面去惊吓他们，然后期待他们能自己弄懂并成为专家。我们的工作是让他们的生活更加轻松。如果我们事先给顾客警告，他们就能更有效地管理信息，而不是在某个星期二早晨被吓一跳。

28.1 为改变而设计

那么你如何为改变而设计呢？

要回答这个问题，我们需要逐步脱离一般情况下对界面的认知，而是去看一看和用户交流的其他方式。在复选框和滑动条控件领域之外，我们使用各种通常情况下和用户体验并无直接联系的事物，例如电子邮件、博客、Newsletter、Screencast，还有屏幕截图。

所有这些都是我们和顾客保持联系的绝佳工具。

为改变而设计的第一步就是评估如何发布更新。

我们是否将在某个明确的时间发布更新？如果不是，能否设定一个？更新会影响你所有的顾客吗？还是只有其中一部分？在更新和那些愿意买单的用户直接见面之前，进行一些小的测试是否会更有好处？

为发布更新设定明确的时间意味着你能给顾客准确的信息，而不是在某天早上突然吓他们一跳。在更新发布给每一个用户之前，先在一小部分人中试用，这将会让你有时间修正"最后关头"的小 Bug。把新功能先告诉那些比其他人更喜欢发现隐藏宝藏的"积极用户"能够获得你最好也最忠诚的顾客的感激，而且让他们感觉自己受到了特殊待遇。

一旦你决定了如何发布，那么就是时候通知他们了。

从一封简单的电子邮件开始，把它发送给每一个注册过的人。那些积极用户会很感兴趣然后告诉自己所有的朋友。普通用户可能也是你最大的用户群体，则会获得足够的预警，知道有些事情马上要发生改变了。那些曾注册过而又废弃了，但又没有删除账号的用户，有时候我们称之为"沉睡者"，则会被提醒而再一次光临你的网站。

而这只是一封简单的电子邮件。就像这样：

> John：
>
> 我们将于 10 月 3 日为 AcmeWidgets.com 做一次重要的更新。
>
> 现在，你可以做下面这些事。
>
> - 和其他人一起分享你上传的图片。
> - 为你看到的图片发表评论。
> - 如果想深入了解某些图片，你还可以和它们的主人联系并交流。
>
> 如果想了解具体情况，欢迎随时过来逛逛：www.acmewidgets.com/ update/tour。

> 这些变动都是自动的，你无需做任何事情。10 月 3 日，和往常一样登录就能开始享受乐趣。
>
> 我们感谢你到现在所做的一切，而且很期待你在新 AcmeWidgets 上的作为！
>
> 你真诚的，
>
> AcmeWidgets 开发团队

通告区

下一步，在网站首页上添加一些关于更新的预告信息。简单又不碍事，但是仍然能够引起注意。

以下是我用过几次的小伎俩，用于各种类型的通告。我称之为，通告区。（我是个天才，不是吗？）

或许你可以在我的个人网站 http://rhjr.net 上找到它，如图 28-1 所示。

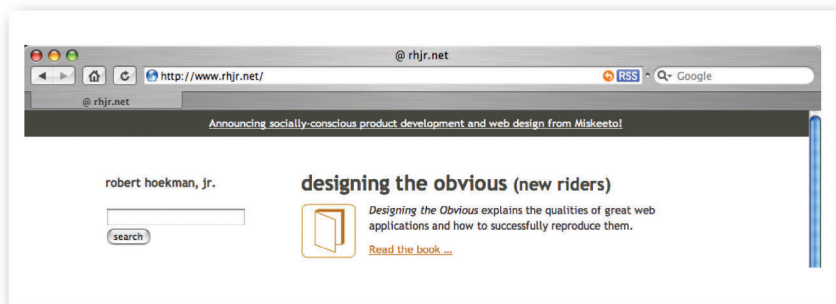


图 28-1

基本上，它是一个沿着页面顶部的黑色横条，里面包含着某个通知，并且通常会链接到另一个信息更详细的页面。

它简单、清晰，而且有效，而这正是我们想要的结果。

使用什么颜色完全取决于你自己，只要确保它能和页面上其他颜色区分开来，因为这样才能引起注意。

28.2　界面元素的再利用资源

Designing the Obvious 一书中，我谈到“欢迎”界面能帮助用户快速熟悉某个产品。同时我也提到在初级用户变成中级用户、不再需要提速协助之后，这些元素仍然可以在新功能或类似情况中再利用。

图 28-2 是我曾用过的例子。不，它没有说“欢迎”。它的目的是为用户介绍页面，只是处于绝大多数情况下“欢迎”文本所处的位置而已。而内容要么是说明性文字，要么是营销广告语。

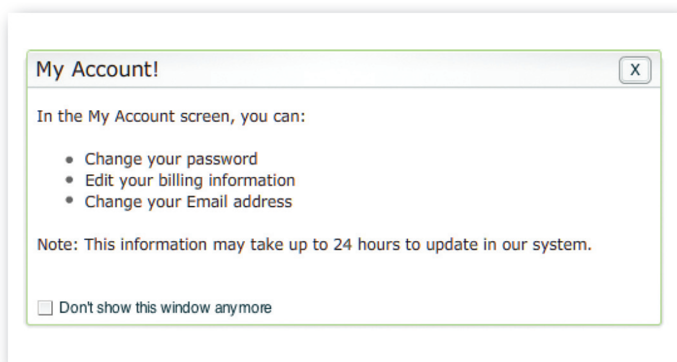


图　28-2

而以下则是如何再次利用它让用户了解即将到来的变动，如图 28-3 所示。

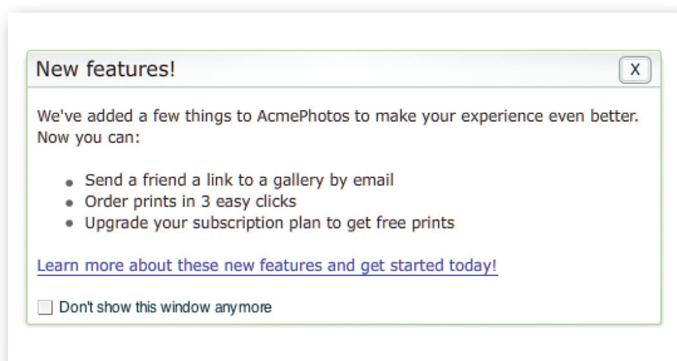


图　28-3

很简单，对吧？用户已经熟悉了这个界面，因为他们在最开始访问的时候就见过它。而再次让它弹出来将会非常引人注意，而且向用户传达出它可能包含了重要信息。

你还需要创建一个页面，包含关于网站更新所有的细节，然后将该页面和这个界面或者通告区进行链接。

为改变而设计的好处远远超过了实现它所付出的时间和精力。这些措施都很简单，但它们都会深刻地影响到在更新发布当天你的网站是否能获得认可。

不管发生什么，不要在某个星期二早晨拿一次天翻地覆的变动去惊吓你的顾客。让他们了解将要发生什么，这样他们才能为变动做好准备，为之兴奋，而且开始再一次地谈论你的产品。

第七部分

离　　开

现实生活中说再见总是很悲伤，但当你面对某个 Web 应用时则不会。很容易就能单击那个魔力非凡的 Log Out 链接（或者 Logout，例如在第 3 章谈到的网站）然后上路离开，而且也很容易一去就再也不回头。

事实情况是，绝大部分人并不是纯粹为了好玩才使用 Web 应用的。他们这样做是为了完成一些事情然后继续回到他们的生活中去。换句话说，大多数人的主要目的是从某个网站离开。实际上，我们之前的每一章都在讨论如何让人们能更快、更有效而且更有自信地完成事情，这样他们才能更快地离开网站。在这里逗留的时间越长，他们的满意程度就会越低。

但是作为设计师，我们的目标难道不是劝说人们逗留的时间再长一点吗？

是的，没错。但是这不能靠让人们陷于复杂的交互和让事情变得困难来实现。它应当靠设计一系列令人愉快的瞬间，让用户对产品感觉良好来实现。所以让用户逗留的最好方法是帮助他们尽可能快而且毫不费力地离开网站，这多少带点讽刺意味。

从页面版式到拖放交互，迄今为止我们已经探讨了如何让事情变得既简单又有效的各种方法。我们鼓励用户探索网站、吸引他们注册、带领他们完成复杂的表单、教导他们使用不熟悉的功能，甚至从长远角度来管理他们的数据信息。

最后一步，我们要探究当他们决定离开网站时会发生什么，不管是暂时还是永久的离开。

本书最后一部分，我们将重溯用户 Sign-out（退出）的过程，讨论如何在人们离开网站一段时间后鼓励他们再次光临。我们还将讨论在他们决定永不再来时，又应当如何处理那些即将永久消失的账户。

离别或许是既甜蜜又哀伤。但对交互设计来说，即使是离别的瞬间，同样也是为顾客形成持久印象的契机。

第 29 章 退出

从 Web 应用中 Sign-out（退出）从来都不是一个复杂的任务，但是有些网站却把它弄得远比需要的更复杂，而且并不是很多网站都能利用 Sign-out 的机会来动员用户再次返回。

通常来说，Sign-out 只是单击网页顶部角落里的某个命令链接而已。

例如在 Blinksale[①]中，你会回到 Sign-in 的页面。这种不错的方法被称作 Closed System（封闭体系），也就是说除非 Sign-in，否则你无法看到该应用的任何部分。而 Srapblog[②]则是另一种情况，单击 Sign Out 后你将会被送回该站点为非登录用户所显示的主页中去。这也是一种很好的解决方案，被称为 Open System（开放体系），在没有 Sign-in 的情况下你仍然能够看到网站的一部分内容。

当然，并不是所有的 Sign-out 过程都这么清晰。

29.1 化简为繁

比方说 LinkedIn[③]的 Sign-out 过程，会让你疑惑是不是自己只进行了一半。单击 Sign Out 链接之后，你将会看到一段文字，大概是如下这个意思：

---

① Blinksale 是一个专业提供在线记账及单据服务的网站。网址为 http://www.blinksale.com/。——译者注

② Srapblog 是一个在线建立多媒体剪贴簿的网站，用户能发布、共享并且能得到打印版本。网址为 http://www.scrapblog.com/。——译者注

③ LinkedIn 是一个方便用户与同事和同学保持联系的网站，也提供了各种寻求工作和商业合作的机会。网址为 http://www.linkedin.com/。——译者注

> **现在您已经 Sign Out 了**
>
> **您的浏览器将会记住您是谁，因此您仍然可以查看您自己的主页。**
>
> **然而，如果想查看或者改变您的私人信息，您需要再次登录。**

这些后面跟着一个标记为 Sign Out Completely（完全退出）的按钮，并且附带说明“当你单击这个按钮时，浏览器将不会记录任何关于你的信息，你必须输入电子邮件地址和密码才能重新使用系统。”

不，我没有开玩笑。

除了极其冗长之外，这些说明文字还会让你感觉自己做错了什么。虽然你已经 Sign-out 了，现在却可能会想是否还得完全地 Sign-out。

我们都倾向于在页面上寻找按钮和链接，同时忽略掉绝大部分说明文字。所以可能很多人看到这个 Sign Out Completely 的按钮都会有些紧张。看到它之后，你可能会觉得自己必须阅读这些说明，好弄清楚到底发生了什么。如果不是因为这个按钮让我们感到自己的安全受到了威胁，可能我们根本不会去理睬那些说明文字。

另一个例子中，我们也能看到不必要的冗长文字。在亚利桑那州某个银行的网站上，当你 Sign-out 时会进入一个页面，里面是这样陈述的：

> **感谢您使用 [Acme] 在线银行系统！您已经成功退出了 [Acme] 系统。**
>
> **如果您希望重新登录 [Acme]，或者回到 [Acme] 网站，请单击下方的 Back to Login 按钮。否则，请关闭您的浏览器窗口以便结束当前对话。**

后面有一个 Back to Login 按钮。

因为登录界面位于网站的主页面上，所以你需要通过主页面进入站点并且在主页面上 Sign-in。但是当你 Sign-out 时，却必须单击某个按钮才能回到主页面，而这又没什么明显的理由。

而且出于某些奇怪的理由，网站的设计师认为你需要用整整两段文字才能理解这个 Back to Login 按钮！更糟的是，他们还假设所有用户都能明白那个神秘的“结束当前对话”是什么意思。

为什么如此简单的事情却被弄得如此复杂？

在这种种混乱之外，只有极少数网站才能做到把 Sign-out 当作一个传达信息或营销的契机。

29.2　动员用户返回

我设计 Sign-out 流程时，总是要么用 Sign-out，要么用 Open-system Sign-out。但在此基础上，我也希望能找到方法鼓励和怂恿用户再度光临网站。

Gmail 在这一点上干得很棒。Sign-out 时，你会回到 Sign-in 页面，但同时你也会看到一些他们希望你能深入了解的功能。在这种情况下，Google 的 Sign-in 页面兼饰两角，既是 Web 应用的首页面，同时又把额外的空间用于行销推广，如图 29-1 所示。

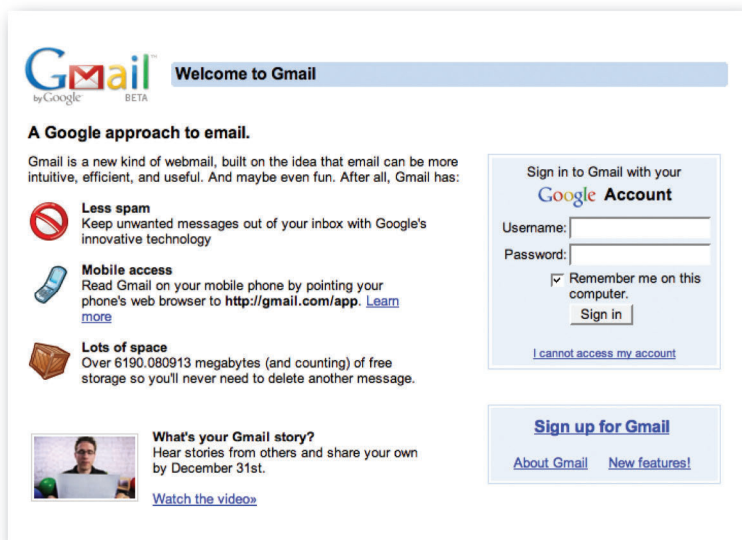


图　29-1

在设计自己的 Closed-system Sign-out 流程时，我窃取了这个想法，而同时也希望能更进一步。这些系统中的 Sign-in/out 页面通常都没有很多内容，因此完全可以把那些空白区域视为用于市场营运的良机。

我由一个最典型的 Sign-in 页面开始，用户 Sign-out 后也将回到这里。它包括用户名和密码的输入域、一个是否记住用户 Log-in 信息的选项，以

及一个找回遗失密码的命令链接。这些元素都被套在一个框体中，在页面居中显示。见图 29-2。

图　29-2

然后，我把这整个玩意向左移动，开辟出一块空间放置行销推广的内容。设计的时候我只是确立了一个模板，无论什么样的信息都可以放在这里，用以描述产品的各项优点。如图 29-3 所示。

图　29-3

29.2.1　空间的再利用

如在第 28 章所讨论的“欢迎”界面那样，Sign-in 页面的推广区域也可以重复使用，例如向顾客提供关于新功能的信息。

为了更详细地说明，我创建了另一个线框图版本，如图 29-4 所示。

图　29-4

为了限制可以放置到区域中的文本数量，我决定把该区域设定得相当小，仅仅和左边的 Sign-in 界面一样大。通过这种办法，不管是为新的或已有功能做广告还是其他什么内容，任何使用该区域的人都必须使用简短的陈述或者项目列表来表现（记住，文本越长，越没人读）。

29.2.2　发布消息

Web 应用的 Sign-out 流程完全可以非常简单，只需点击一次然后退回到 Sign-in 页面即可。但如果加入一点创造力，这个本来很空的页面也可以转变成一次市场营销的机会。

它可以通过指出用户尚未发现的功能鼓励用户再次回到站点，或者引导用户了解新功能，甚至可以宣布网站新取得的成就，或者其他影响到整个用户群体的新闻，例如发布例行维护通知等。

而正因为这个页面除此之外再无其他任何东西，因此与 Web 应用中其他拥挤的页面相比，文本会更加吸引人们的注意。

第 30 章
抹去那些尘封的用户

尘封用户（Dusty User），或者如 37signals 在 Signal vs. Noise（信号与噪音）中的文章 Waking up the sleepers（唤醒沉睡者）[①]里所称呼的“沉睡者”，指的是那些注册了某个 Web 应用或者服务，却又放弃了账户的人。这种情况很有可能发生，例如用户对某个产品厌倦了，或者找到了一个更喜欢的替代品，甚至仅仅只是因为他们把这个应用给忘了。

到现在为止，我可能已经是上百个 Web 应用的“尘封用户”了，我相信你也一样。

那么作为设计者，我们应该如何应对这种局面呢？事实上我们的目标是让用户保持兴趣和热情，那么我们应该如何让这些沉睡者复苏并再度活跃起来呢？

这是一个我很想回答的问题，但通常的思路认为，这种设计工作都是市场部门的职责。广泛地宣传、在电视和网络上做广告、提供优惠或者其他折扣，这些都是传统的宣传手段，人们认为它们足以让那些“尘封用户”行动起来。

但是，尽管这些营销手段在保持品牌知名度方面做得很优秀（或者很差劲），它们却很少能真正地鼓舞用户。吸引我们注意的那些东西其实非常个人化。它们不是路边广告牌上不停喧嚣的宣传妙句，而是一些私人信息，是针对于我们的需求而建立的诚恳联系。

① Signal vs. Noise 是 37signals 的一个博客，主要内容包括设计、用户体验、文化等。Waking up the sleepers 是其中一篇日志，具体网址为 http://www.37signals.com/ svn/posts/611-waking-up-the-sleepers。——译者注

换句话说，唤醒这些沉睡者绝不仅仅是市场部门的工作，它同样也是设计师的职责。毕竟，和顾客沟通并且建立联系，在这一点上我们做得最好。

为了更好地描述这个观点，我设计了一个解决方案清单，对象是一个虚拟的项目，名为 Acme Photos（Acme 相册）。

30.1　变怠惰为热情

我们抹去“尘封用户”所需要的工具，有很多其实早就已经存在，而且在其他某些情况下我们已经使用过了。

比如说，我们很多人都已经发送过 Newsletter（邮件时事通讯）。尽管通常它们都被用于公司的最新报道或者描述新功能，但也同样能够提醒“尘封用户”现在有个好理由回到网站并开始再度使用它。

为了让这种效果最大化，我们可以在 Newsletter 中加上链接进一步鼓励“尘封用户”进行探索。此外还可以提供折扣或者其他因素来吸引这些用户。以下有一个例子：

> 有一段时间没有使用 Acme 相册了？
>
> 让我们使你相信这值得瞧一瞧！我们已经创建了一系列短小而有趣的视频教程，展示了一些很棒的新特性。而且它们都在一个地方，所以你能很快上手并且了解我们现在能为您做些什么。
>
> 了解我们的新特性！
>
> 如果这还不够，我们还愿意为你提供特殊优待。如果现在距你上一次登录账户已经超过了 1 个月，我们将在 2 月底之前为你购买的所有商品提供 50% 的折扣！
>
> 只需在付款时输入以下推广码：
>
> FEB_PROMO
>
> 开始购物！

明确地针对“尘封用户”的需求，同时为他们提供足够的动机，我们将吸引那些摇摆不定的人们发现网站的价值，为自己的站点提供第二次机会。

30.1.1 个性化沟通

另一个选择是发送个性化信件。例如，我最近收到了一封传统信件，来自一个信用卡公司，我有一阵子没使用那个账户了。它用一种幽默的语调简单地询问："咱们最近干了些什么？还是什么也没干？"

这封信件的立意在于，它让我知道该公司已经注意到我有一段时间没有使用自己的信用卡，并且鼓励我再次用这张卡消费。他们在信中为自己没能让这张卡体现出价值而道歉，而且提供了一些升级举措，以确保我能从信用卡中得到确实想要的东西，例如持卡消费能获得的旅游及现金回馈、信用额度的提升，以及在线付款功能。同时他们还声明只要我能再次成为活跃的信用卡用户，他们愿意竭尽所能满足我的需求。

我真的非常感激他们，特别是因为我正在开始 Miskeeto[①]计划，需要一张信用卡为公司开支。我不仅打电话为现有这张卡办理了升级，还同时申请了另一张商务信用卡。从那时候起，我开始有规律地使用这两张卡，成为该公司一个良好的消费者。而所有这一切都是因为他们给我的一封信吸引了我的注意。

以下是转换成电子邮件形式的一个例子。

> 亲爱的 Maggie：
>
> 我们注意到你有一段时间没有使用 Acme 相册了，于是怀疑自己做错了什么。你是不是遇到其他什么人了？你不再爱我们了吗？
>
> 我们需要你再给一次机会！
>
> 为了证明自己不变的真心，我们已经创建了一系列短小而有趣的视频教程，展示了一些很棒的新特性。而且它们都在一个地方，所以你能很快上手并且了解我们现在能为您做些什么。
>
> 了解我们的新特性！
>
> 如果这还不够，我们还愿意为你提供特殊优待。如果现在距你上一次登录账户已经超过了 1 个月，我们将在 2 月底之前为你购买的所有商品提供 50% 的折扣！

---

① 见第 15 章注。——译者注

> 只需在付款时输入以下推广码：
>
> FEB_PROMO
>
> 开始购物！
>
> 我们渴望让你成为一位开心的顾客。如果这封邮件没有成功，请让我们知道还能为你做些什么。通过以下链接向我们发条消息，你将能和真人接触并且进行真实的交谈，让我们知道如何才能赢回你的心。
>
> 和 Acme 相册的职员交谈！
>
> 我们希望很快能再次见到你。真的。我们想念你。
>
> 你真诚的，
>
> Acme 相册团队

就像你所看到的，我在这封电子邮件里站在一种失恋的角度。诙谐和幽默对于抓住用户注意力来说相当有帮助。

最重要的是，我们发出去的讯息应该富有人情味，而不是呆板的、充斥着行业辞令和广告语气的营销套话。这类讯息应当是一种会话般的口吻，完全聚焦于接受方，而不是我们自己。

当然，由于这两个例子都包含了通向视频教程的链接，我方案清单中下一项就是要创建一堆短小精湛的 Screencast。在第 8 章我们谈到过如何使用视频作为辅助，所以这里我不会再涉及那些细节。

这些东西应该能有帮助，但我们怎样才能弄清楚这些用户当初被尘封的原因呢？

30.1.2 调查

首先，我们可以问他们为什么离开这么久。为了实现这一点，我们可以在解决方案清单里面加上调查。

使用例如 SurveyMonkey[1]这类的服务，我们可以轻松地开展客户对 Acme 相册感觉如何的小调查，并和优惠或其他奖励项目一起发送给“尘封用户”。

这种调查的完成率毋庸置疑会很低。但即使只有很小一部分人回应，我们也可能从中学到很多东西，比如可以了解在应用中还差什么，或者是哪里让这些用户失望。

通过调查，我们能够看到用户的各种倾向，并根据反馈考虑如何改变设计。

但调查并不是最后一步棋。形象一点来说，我们也可以和用户坐下来边喝边聊。

30.1.3 对话的机会

正如 *Cluetrain Manifesto*[2]一书的作者所说的，“市场即会话”。但只有很少的公司真正去聆听他们的客户以及从事真正的会话。

为了让这种会话开展下去，我们可以为“尘封用户”提供一个机会与真正的开发团队成员交谈，并把他们的关注点和愿望描述出来。这也可以通过电子邮件形式实现，因此重要的只是找到方法让它更容易一些。

这可以很简单，例如创建一个电子邮件地址，并承诺用户发送的每一条评论都会得到真人回应。但我们还可以走得更远。例如，我们可以邀请那些“尘封用户”成立一个小组，目标聚焦于产品日后的改进工作。

这些用户可以参与论坛或者讨论列表，和内部员工一起讨论他们希望看到的变化。为了补偿他们付出的时间和精力，我们可以提供商品折扣作为参与活动的酬劳。

---

① SurveyMonkey 是一个在线问卷调查网站，提供各种问卷模板，可定制性较高，还能将调查结果以 PDF 格式存档。用户只需将问卷的地址以链接形式添加到网站、邮件等处，就能通知被调查者填写。而且只要调查对象低于 100 就无需支付费用。同时也支持中文。网址为 http://www.surveymonkey.com/。——译者注

② *Cluetrain Manifesto*（线车宣言），Weinberger、Searls 和 Levine 于 1999 年合著。中文版名为《市场就是谈话：影响传统企业思维的 95 个观点》，中信出版社出版。有人认为该书实际上是 Web 2.0 浪潮的先知，甚至其作者也可归入信息资本主义的伟大预言者行列。该书网站为 http://www.cluetrain.com/，可免费阅读原版内容。——译者注

为了创造一种“排外”的感觉，我们可以限制小组的编制，例如，100 位用户。我们可以在向所有“尘封用户”发送的电子邮件邀请中解释小组的细节，然后告诉他们只有 100 个人将会被选中。

排外感将会激发兴趣。把小组限制为 100 个人将会让用户感到自己正在参与很特殊的活动。同时这也会把小组的规模控制在可管理的范围内，长远来说会对公司真正起作用。而另一方面，那些未获选加入小组的用户也会认识到公司正在采取措施以图打动消费者。

30.1.4 不要做得太过火

考虑了上述一切之后，成功的关键在于不要做得太过火。

如果某个用户只是想独自呆着，我们可能很容易就会因为过多的打扰而冒犯他，甚至会迫使他注销原本只是放任不管的账户。与此同时我们也不希望让人有抓着稻草不放的感觉，所以这里需要制定一些规则。

当某个账号被尘封了相对很长一段时间之后，我们才应该发送第一封通知他们查看新功能的电子邮件。如果是那种通常需要每天使用的产品，电子邮件应当发送给那些至少 1 个月没有 Sign-in 的用户。而像我们的 Acme 相册这种平均每周登录一次的产品，电子邮件应当发送给 3 个月没有 Sign-in 的用户。

这里并没有什么数学公式，而只是根据使用频率推算的一个大概值。

而下一步，在第一封电子邮件发送一至两周之后，再发送一封更个人化的电子邮件，比如刚讲到的那个例子。这个额外的讯息将会进一步鼓励用户，因为第一封电子邮件还记忆犹新。如果某位用户到此时还是什么也不做，我们最好就让他一个人呆着吧。

至于“尘封用户”们构成的小组，明显地一年只能组建一到两次，每个小组可以持续数个月。

关于小组的电子邮件可以说明将会为期 3 到 4 个月，而当本期结束后将会再组建新的一个小组。同时也可以说明成员并没有义务必须得表现活跃，但是那些频繁的参与者将会获得更多的奖励，例如更低的折扣等。

如果每年可以组建并维持 2 次这种小组，邀请的电子邮件应当在第一次讨论之前两周左右时间发送出去。那些接受邀请的人需要这段时间来重

新熟悉产品，甚至还有可能向自己的朋友提及这个公司能做这么一件事，给人的感觉棒极了。

通过折扣、幽默诙谐的讯息，以及提供和真正开发团队成员关于产品进行交流的机会，那些“尘封用户”将会有足够的契机对产品重新产生兴趣。

最后，所有这些都是有效的客户服务行为。每一个设计的决定、每一次会话、每一篇博客日志、评论、电子邮件、电话等，它们都是为了提供良好的客户服务。

做到这些，你就能收获忠诚。

第 31 章 由他们去吧

令人悲哀的是，并不是每个人都热爱我们的劳动成果，不管它有多好。我知道这一点令人不快，但它是事实。哪怕我们做了自己所能做的一切来劝服用户注册，设计出不言而喻的交互和说明来教导他们使用产品，为产品提供大量的价值，并且让数据管理尽可能地轻松简单，甚至尽力消除那些尘封的用户。但还是有很多时候，客户离开了就再也不回头。

尽管这是一个悲伤的时刻，就像耻辱柱上的标记，但我们仍然无法知道它到底何时会发生。并没有实时的报警器能告诉我们某人在什么时候会删除自己的账户，我们也无法趴在他们的肩头目击导致这一宿命般瞬间的体验到底是什么。相反的，它的发生超越了我们的知识。我们所能看到的只是那些为了衡量自己进展而收集的数据，而它们把这些我们赖以生计的客户简化成了一系列数字和图表。

就像在 Web 应用里的其他所有东西一样，我们挽留客户的最后防线仍然需要自动化。只有这样系统才能帮我们工作，我们才能有时间呆在家里看看 *The Daily Show with Jon Stewart*[①]之类的电视节目。

而最重要的是，我们需要优雅得体地应付这种状况。这听上去似乎有些违反常理，然而遗弃一个账户本来就应该像其他任何事情一样简单容易。由于每一款应用程序面临的挑战都各有不同，而且它们的目的和领域也不尽相同，所以我认为与其只讲一个单独的故事，还不如针对自己曾经面对的各种情况举一些实例要来得更有效果。

① *The Daily Show with Jon Stewart*（乔恩每日脱口秀）简称 The Daily Show，每周一至周四晚 11 点首播。Daily Show 不同于传统的 BBC News 或 CBS Fox CNN 的每日新闻栏目，而是从搞笑的角度演绎每日国内外发生的重大新闻。每期除了邀请嘉宾做访谈，还有很多有趣的单元，整个节目笑声不断。该栏目网址为 http://www.thedailyshow. com/。——译者注

31.1 优雅地失败

不管是何种情况，我都尽量避免在任何时候让客户感到自己受到了限制。我不知道是否有什么理论可以验证我的这种说法，但是我非常确信这是一件正确的事。

想想看。

感觉自己受困于某项服务或协议的用户会感到自己受到了限制，而理由可能仅仅只是因为没有有效的方法将注意力转到其他东西上去。一旦这种事情发生，他就会愤慨，而愤慨是让某人说你坏话极为有效的方法之一。同样地，如果一个人在任何时候发觉自己即将受到限制，他绝不会待更长时间，反而会马上逃脱。这种困住用户的战略可能对手机服务供应商来说有效果，但对网络来说绝无可能。

所以，即使我们在挽留客户的战役中失败了，也要在这个过程中保持我们的尊严。这样不仅能避免让我们看上去过于绝望，同时还能抓住最后一个机会让客户对我们留下好的印象。

以下是我的建议。

31.1.1 打点行李

首先，让客户能为自己打点行李，这点非常重要。例如某个用于管理个人数据的应用程序，必须为日后可能的注销行为提供一种可靠的保留信息的方式，否则客户在一开始根本就不会感兴趣。

方便用户集中组织小型项目以及搜集内容的 Backpack[①]（37signals 开发）在这方面就做得很好，它的账户页面提供了 Export Your Data（导出你的数据）的功能，如图 31-1 所示。只需单击这个链接，你的 Backpack 页面所有的内容就会生成一个 XML 文件并发送到你的电子邮件中去。该功能就处于极易访问的账户页面中，也能够用来提升或缩减自己的订阅计划，或者设置其他账户信息。这意味着你无需任何努力就能取消自己的账户。

① Backpack（背包）是一款信息组织、日历、工作安排的在线软件。其简洁的应用界面能帮助用户管理手中的项目，还能从许许多多的有用资源那里抓取自己的所需要的数据。该软件网址为 http://www.backpackit.com/。——译者注

没有隐藏页面，没有复杂的步骤，只要在任何一个 Backpack 页面上进行一系列单击操作，你就能导出所有的数据然后取消自己的账户。

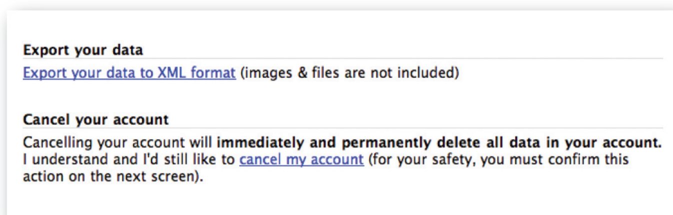


图 31-1

知道很容易取消 Backpack 账户这一事实，会让人们在一开始就愿意接触这款应用程序。由于转换所需的代价（选择其他应用用户所需花费的时间和金钱）非常低廉，用户从开始使用产品时就感觉很舒适。

Google Reader 是一款博客阅读器程序，能让你导出自己收藏的所有 RSS-feed 的 URL，如图 31-2 所示。你可以把它们导入到另一款程序中，从中断处重新拾起一切。很多博客阅读器程序都支持导入这种列表，所以你完全可以在一天当中尝试各种阅读器，逐一试用直到找到那个最适合你的为止。

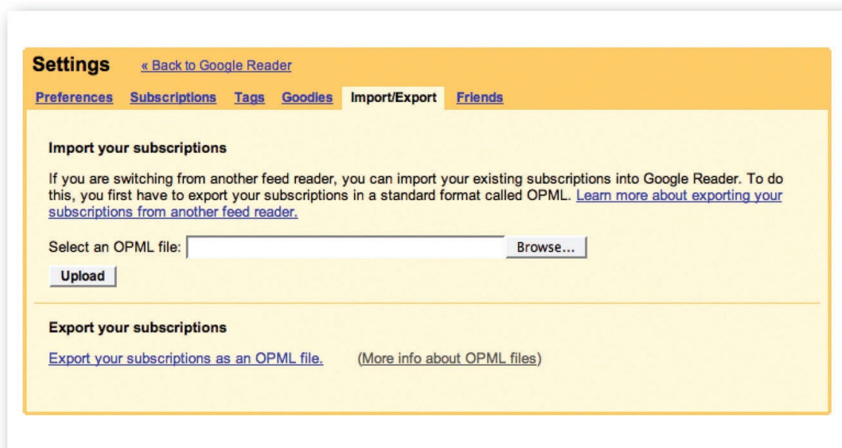


图 31-2

而在文件在线存储系统 DropSend[①]中，在你取消账户之前所需要做的只是把所有存储的文件全部下载到自己的计算机里即可。

31.1.2 离去却（也许）并未失去

而在上述这些例子中，用户取消账户的行为本身也非常简单。

在 Backpack 里，你只需点击 3 次就能取消一个账户：Account（账户）链接、Cancel My Account（取消我的账户）链接，以及 Yes, permanently cancel and delete my account（是的，永久取消并删除我的账户）确认按钮。

点击 3 次，你就完事了，再也没有比这更简单的了。当然，除了 Google Reader。

在 Google Reader 里你无需做任何事。因为 Google 提供了一种 Single-Sign-On（一站式登录，简称 SSO）的功能，让你能用同一个用户名和密码登录所有的 Google 服务。因此不管你使用与否，Google Reader 总是处于开放状态，并且 Google Calendar（Google 日历）、Google Documents（Google 文件）和 Gmail 都是如此。如果想使用另一款阅读器程序，例如 Bloglines[②]，仅仅只需把你的订阅地址导出为 OPML（Outline Processor Markup Language）[③]文件，然后导入到 Bloglines 里即可。

而在 DropSemd 里，也只需单击 My Account（我的账户）标签页、Please Cancel My Account（请取消我的账户）链接，然后单击确认链接 Cancel my account and delete all files（取消我的账户并删除所有文件）。

不管是什么应用程序，客户离开都应该是一件非常简单的事。一直以来，你为了用户在产品中有一个完美的体验而尽了最大的努力，而最后却想欺骗或者强迫客户违背自己的意愿留在这里，这对你没有帮助，对客户也没有帮助，只会导致最坏的口头宣传效应。客户会在他们的博客里说

---

① DropSend 能让用户发送 1G 容量的超大尺寸文件（普通 E-mail 附件限于 10M 以内），并提供文件在线备份、共享等功能。但更高级的服务需要收费。网址为 http://dropsend.com/。——译者注

② 在 Google Reader 推出以前，Bloglines 可以说是最好、使用人数最多的 RSS 阅读器，但随后一直未能超越 Google Reader。于 2005 年被 Ask Jeeves 收购。网址为 http://www.bloglines.com/。——译者注

③ OPML 即“大纲处理标记语言”，也是一种 XML 规范的文件格式。它提供了一种批量 RSS 导入导出功能，即可以把多个 RSS 订阅地址集中到一个 XML 文件里面。——译者注

你的坏话，并且劝告其他人离你越远越好。

不过，你仍然可以做最后一次尝试，劝阻客户的离开。

DropSend 的账户取消程序（当然只针对免费账户）会引向一个页面，里面这样说到：

> **你的 DropSend 账户不会花费你的任何成本……**
>
> **你确定真的想取消吗？这是 DropSend 的免费计划，你无需任何成本。为什么不保留你的账户，以防有一天需要发送某个大文件？或者需要再次往 DropSend 里备份什么东西？**
>
> **如果你的账户被取消，你所有的文件都会立刻永久性地删除。这项操作无法撤销！**
>
> **取消我的账户并删除所有文件**
>
> **回到我的账户**

这段话非常客气，是一种友好的语调，而且可能在一段时间内确实会收到效果。就如同它所指出的，一个拥有免费账户的人确实没有理由取消它，他完全可以轻松地保留账户以防日后需要。

另一个选择是在用户取消账户时发送一封最后的电子邮件。里面可以包括这样的内容：

> **看到你离去我们感到非常遗憾！我们希望你在 Acme 相册中获得了良好的体验，而且希望我们没有为你带来任何方式的冒犯或不便。**
>
> **如果你愿意给我们最后一次机会，我们会非常乐意告诉你一些我们的新特性，并且为你的所有订购提供折扣。**
>
> **了解新特性并且接受折扣！**
>
> **但如果你真的准备离开我们，希望你能花一点时间告诉我们为什么。这对我们改进站点将有莫大的帮助，也许以后还能再次夺回你的心。**
>
> **告诉我们哪里出了问题。**

> **感谢你一直以来所做的一切。我们非常感激，而且再次地，很遗憾看到你离去。真心祝你好运。**

不用那种绝望般的满腹牢骚，我们可以最后一次请求正要开门上路的客户多逗留一段时间并再给我们一次机会，而这并不会失去我们的尊严。

优雅的失败对于 Web 应用来说是非常容易做到，而我们留给客户的正面印象将远比失去客户所带来的痛苦要来得长久。如果应对得体，哪怕是一个并不满意的客户也能为我们带来积极的反馈，甚至是口头上的宣传。

而这所需要的只是在客户离开时多表现出一点敏锐、尊重和专业。

优秀设计的金钥匙

在 *Designing the Obvious* 一书中，我描述了设计一款伟大的 Web 应用所需的 7 条核心原则，以及如何把它们运用到实际项目中去。我说，要想设计伟大的应用程序，我们需要：

- 只提供用户所需
- 让初学者尽快转变为中级用户
- 尽可能地预防错误，如出错也应友善处理
- 精简和优化交互与任务流程，让最复杂的软件清晰易懂
- 为支持特定行为而设计
- 持续地改进和优化我们的流程与程序
- 忽略用户的某些要求，坚信自己的理想和洞察力

这些原则引导着我进行了一个又一个的项目，证明它们确实是值得依靠的一系列基本指导方针。通过这些指导方针，我们完全可以实现极为易用的设计，让用户只需常识便能有效地使用我们的产品。

而在这些宏观原则之外，对于 Web 应用来说，每一位用户的体验都由一连串单独的瞬间构成。这些瞬间对用户与软件的联系和感知都会产生重大的影响，而其中的每一个瞬间都需要精心设计。为了让这些瞬间尽可能地完美，我们也需要为每天面对的各种具体的独特的设计问题找寻解决办法。所以在本书中，我们探究了 30 多个微观原则，它们能帮助你解决从页面布局到社区网络交互中可能发生的每一个问题。

而在所有这些下面隐藏着一个简单而又基本的真理。我在本书和 *Designing the Obvious* 里所讨论的每一个原则、理念和想法都深深植根于这个必要条件，它是任何设计项目的绝对核心：

设计即传达。要想设计出伟大的体验，我们必须成为伟大的传达者。

传达其他所有一切

贯穿本书始终，我们一直着眼于改善各个瞬间。从第一印象到最终印象，从拖放操作到发布更新，它们构成了用户使用 Web 应用所得到的最终体验。我们探究了帮助人类与 Web 应用交互的各种原则和理念。我们帮助他们分析、判断、从过失中恢复，甚至在他们改变主意或者离去之后仍然保持关注。我们帮助他们获得导向、形成思路、加深记忆、养成习惯、产生信赖、获得灵感，并且感觉自己有成效。

作为设计师，我们在设计每一个交互时都会面临各种挑战。我们需要知道人们在网络中是如何交互的，是什么让他们热爱或憎恨一款产品。我们需要知道何时何处应该使用何种设计思路、了解最好的实践方法和流行趋势、学习如何更好地表述。而最重要的，我们还需要知道如何让一款软件的每一个细节都容易理解，这样用户才能实现他们的目标，而相对的，我们也能实现自己的目标。

为了接受挑战，设计师在大量的设计方法和理论基础上加以各种变化和改进，并且提出了无数的设计文档、流程图、图表以及其他所有你能想象到的资料，力图控制最终的结果。

然而，我们并没有魔法子弹能自动命中目标。想让我们的作品比其他的更好，这其中也没有捷径。没有哪个方法可以一直遵循，没有完美的设计文档，也没有可以随时采集天才想法的魔树。为了调整和精炼自己的能力以达到成功，我们在前进的过程中可以一百万次地改变设计方法、变换使用的文件类型，或者任意地进行其他方面的改变。

而最终，它们一点都不重要。如何才能得到完美的设计一点也不重要，因为这是一个可以重复的过程。正如刚才所说，为了调整和精炼自己的能力以达到成功，我们在前进的过程中可以一百万次地改变设计方法、变换使用的文件类型，或者任意地进行其他方面的改变。

但不管怎样做，我们首要的任务是传达，而且这一点从未改变过。

我们是其他所有一切的传达者。不管是设计应用程序、演示幻灯片、设计文档，还是餐馆里的酒单，我们的目标永远是向用户传达它们是如何

工作的，而他们又能从中得到什么，以及为什么它们很重要。

为了做到这些，我们通过各种设计手法、指示性元素以及其他手段来传达一款应用程序及其交互行为的目的、利益和使用方法。我们通过颜色、图像和文字来传达情绪。我们通过客户服务和高质量的产品来传达对用户的奉献精神。

不管意识到与否，我们无时无刻都在传达各种各样的事物。而如果我们传达得不错，就能收获尊重和忠诚。

然而，如果我们传达得很糟糕，就会收获相反的东西。我们拿出“不是很优秀”的设计时，等于是在向用户传达我们要么“不是很有能力”，要么“不是很用心”。

作为设计师，我们拿出的设计必须能告诉用户，我们正在百分之百地用心让他们快乐，正在百分之百地用心确保他们能在我们的产品中逗留，并且能持续地发现价值。

应用本书以及 *Designing the Obvious* 中所讨论的原则，将在相当程度上帮助你做到这一点。

最后我们必须澄清这样一个事实：所谓“做出伟大的设计都很艰苦”的观点是一个谬论。这条谬论说良好的传达都很难以实现，还说所谓的简单其实是把复杂从用户交互这一前端转移到了后端。

做正确其实更容易

似乎有点违背常理，与错误的讯息相比，传达正确的讯息所需要的思考或精力反而会更少。

让我解释一下。

我 19 岁时陪一位朋友参加了一个古典吉他的大师培训班。在这种培训班中，经验丰富的大师级音乐家会对学生的弹奏进行指正，并且提供建议帮助他们进步。

在我上的这个培训班里面，大师是一位经验丰富而且很著名的音乐家，弹奏得非常棒。他的吉他似乎能吟唱出音乐里每一个音符所包含的情感细节，令人惊异。而我的朋友无疑对自己可能会受到批评而有一些不

安。

不安转变成他手指反常的紧张。

大师注意到了这一点。而当他指正我的朋友时，说的话我永远也无法忘记。他说：

> **“吉他弹得不好很难，弹好则非常容易。”**

这句话让我非常困惑。几年来我一直不知道他的意思到底是什么。他就像一个巫师一样告诉了我一些可能了解的最伟大的秘密，却留给我自己去琢磨其中的意思。

但是最后，因为自己也打鼓，我逐渐开始理解他想说的了。

如果一位吉他演奏者弹得很好，那是因为他到达了一种类似禅的平静状态。一种融入的感觉。一种高度的集中。他的手腕非常舒展，他的呼吸非常平稳，他的手指非常放松，并且准备从吉他指板的一处流动到另一处。他并不是在弹奏音乐，而是在音乐中畅游。那一刻他的整个灵魂只是创造的音乐里面的一部分。这是一种非常美妙的体验，对弹奏者和听众来说都是如此。

作为这种完美、平静的控制的结果，吉他弹奏者能通过更少做到更多。并不是勉强在每一个小节中塞满音符，他能让音乐呼吸。并不是一直试图证明自己的灵巧和高超的技术，他知道那些没有弹奏的音符其实和弹奏出来的音符一样重要。

他并不会把自己所知道的一切都强塞到音乐里。他拿走的和留下的一样多，甚至更多，因为在音乐里，空白能带来生命。

打鼓也是一样。还有绘画、篮球，还有 Web 应用（以及其他很多东西）。

用几句话来描述一段说明文字，远比写上整整一大段要容易。在应用程序里只提供必需的内容，远比把我们能想到的一切细枝末节都添加进去要容易。在页面中留下一些空白，也远比把每个角落都塞得满满的要容易。

换句话说，省去一些东西，远比把它们全塞进来要容易。

人们使用这样的应用程序，远比使用那些臃肿的程序要容易。多余的特

性、复杂的交互还有时髦的界面小工具，它们更多地是在炫耀开发者的技术，而不是服务于实际用途。

今天，在网络上有许多关于可用性的研究资料、大量关于设计方法和指导方针的信息，我们还有各种机会通过网络社区和电子邮件获取专家建议。网络的历史发展到了现阶段，我们无法再为差劲的设计找到理由了。

但最重要的是，存在着一个简单的事实，就像我那位弹吉他的朋友那样：做好比做坏要更加容易。

要想取得成功，你只需要留意自己要传达的讯息，并且充满热情地去好好传达它们。结果其他所有一切都会纷至沓来。

通过留意所要传达的讯息，你的目标将变成想办法如何只花最少的努力去完成传达。通常来说，你做得最少的那种方法其实就是最正确的解决方案。

我希望在为瞬间而设计里面所讲述的这些故事能帮助你做到这些。

最后的补充

我们每个人都必须开辟自己的道路，去领会是什么让我们成为彼此不同的个体。必须了解自己的局限、力量、独特的癖好以及激情等。必须注意我们是如何思考和行动的、如何与同事交往，以及如何应对不利情况。必须注意我们是如何设计，并且找到更好的办法来改进，不管这些方法是来自于研究、人工观测还是大量的实践。还必须考虑我们的环境、工作所使用的工具，还要了解这些东西是如何影响我们的能力。最后还必须留意我们生活的各个方面，这样才能为不同的个体进行设计。

不管你怎么做到这些，最重要的是在生活、工作以及设计中你一直留意着。这样成功才会接踵而来。

再重申一遍，好的设计是由一大堆差的设计演变进化的结果。有些时候，你第一次尝试就能有好运气，但大多数时候则不能。这是一条漫漫长路，只要你坚持走下去最终一定会发现自己一直在寻找的东西。无论如何，继续前进。

这才是我最重要的建议。

跟随你自己的福祉。找到你自己的禅。设计属于你自己的瞬间。

* * * * *

如果希望继续阅读我关于设计的思想，请访问我的博客：www.rhjr.net/theblog。如果希望深入了解我的公司 Miskeeto，请访问 www.miskeeto.com。